Web Services: Software-as-a-Service, Communication, and Beyond

Wu Chou
IEEE Fellow, Ph.D.
Director, Avaya Labs Research
Avaya USA

September 23, 2008
@ICWS 2008, Beijing China
Agenda

- Cloud computing - how deep are we in the cloud?
- SaaS (Software-as-a-Service) - value, challenge, and what follows
- Web services in communication – a paradigm shift to make communication-as-a-service
- SaaS (mobile) endpoints, e.g. WIPdroid, for distributed services computing
- Summary
Cloud Computing

Revolutionizing the services computing environment

- Unique value proposition
  - Next generation Internet based computing and next generation data centers
  - Scalable and massive on-demand computing power/resources
  - Reusable and make computing as a service

- Technical advances
  - Virtualization and hypervisors: Xen, VMware, Hyper-V, etc.
  - Algorithms and software: MapReduce, Hadoop, etc.
  - CPU and OS architectures, e.g. Intel VT instruction set, VMX mode, etc., for HVM, and hybrid HVM/Para virtualization
  - Fast CPU, fast networks, lower hardware (cpu, memory) cost.

- Strong industry/government support
  - Google/IBM academic Initiative, NSF, etc.
  - Green enterprise, and energy efficient data centers, etc.
Cloud Computing – Business Model

- **Capture the long tail in computing**
  - Large scale computing needs may occur sparsely, e.g. certain trading application runs only when the Market is hectic above certain level
  - May not be able to scale up with the dynamic load variations if applications are locked to dedicated hardware/servers
  - It can be a huge oversale, in terms of cost, usage, energy, etc.

- **Cloud computing is on a fast-track**
  - Public Cloud (pay by usage): Amazon EC2/S3, Google App Engine, Force.com, 10gen, GigaSpaces, 3tera, etc. with more to come.
  - Private Cloud (used internally but cross various IT and business needs)
    - Moving applications from dedicated hardware box to distributed data centers
    - Scalable with value to large, medium, small enterprises and even individual uses, if technical and service quality issues can be addressed properly
What is “Cloud Computing” - something from an ancient story

- Blind men and an elephant (my version)
  - A King ordered his six blind wise men to touch an elephant to tell him what it is. (why?)
  - He got the following answers: pole (leg), rope (tail), tree branch (trunk), a hand fan (ear), wall (belly), solid pipe (tusk)
  - All are right but everyone has a different answer (each one of them touches a different part of the elephant, and an elephant has all features mentioned.)
  - The King realized that each of his wise men is brilliantly smart and yet limited by what they touch. He needs to keep all of them!

“Blind-monks examine an elephant” , an 1888 ukiyo-e print by Hanabusa Itchō.

- The King also realizes: he can be a super wise man if he can have all their wisdoms (perhaps just by marshups/extend them to form a more thorough and holistic view).
What is cloud computing

Six related (or confusing) views with cloud computing

- “Grid Computing”
- “Service Hosting”
- “Utility Computing”
- “Virtualization plus data automation or autonomous computing”
- “SaaS (Software-as-a-Service)”
- “Web 2.0, Web 3.0 or Web x.0”

- “Cloud Computing” (a.k.a. the “Cloud”) has all features mentioned, and yet not limited to any one of them
- The “Cloud” is fast evolving (great opportunity for research)
- We are deeply in the “Cloud”, and we are limited by things we touch!
A practical view of the cloud computing

Enterprise Data Center
- a practical view of using cloud computing in data center

Infrastructure services framework of a new enterprise data center

Workloads
- Innovation Enablement
- Software Development
- Virtual Classroom
- Web 2.0 Data Intensive Processing
- Scalable Transaction Processing

Management
- Dynamic Provisioning
- Dynamic Scheduling
- Self-Service Portal
- Monitoring
- SLA Capacity Planning

Virtualization
- Virtual Servers
- Virtual Storage
- Virtual Networks

Physical Hardware
- IBM System z, p, x, BladeCenter
- Storage
- Networking

“From Cloud Computing to the new enterprise Data Center”

© 2008 Avaya Inc. All rights reserved.
SaaS – Software-as-a-Service

Distinct value position

- Capture the long tail of software applications
  - Deliver software applications, host and operate the applications as service (e.g. Salesforce.com)
  - Do not pay for the software, the equipments, the software installation and support, etc. Instead, user is charged by usages.

- *Being vertical without being physically vertical*
  - Enterprises can only be vertical in certain areas due to the limitation on resources, history, work force, and knowledge base.
  - SaaS providers from vertical domains can make an enterprise vertical, without being physically vertical through many years of investments, e.g. sales force management, human resource management, etc.

*There is a natural binding between SaaS and cloud computing, but they are not equivalent.*
Enterprise Software Trends in 2008

**Question:** Please select the most important trend impacting your business.

‘Most important’ in 2008*
Percent, n = 857

<table>
<thead>
<tr>
<th>Trend</th>
<th>2006</th>
<th>2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>SaaS/SaaS platform</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>Web services/SOA</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>Open source</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>Offshoring/globalization</td>
<td>15</td>
<td>13</td>
</tr>
<tr>
<td>Software industry consolidation</td>
<td>17</td>
<td>7</td>
</tr>
<tr>
<td>Others</td>
<td>4</td>
<td>16</td>
</tr>
</tbody>
</table>

* SaaS platform was not explicitly asked in 2006

Note: Percentage budget allocated to software rose from 30% in 2006 to 32% in 2008, and expected increase to 35 percent in 2010.
SaaS – Software-as-a-Service

• Multi-tenancy
  – Single instance multiple tenant, data/application isolation, tenant data security, customization, tenant management and billing

• SLA and Service Integration
  – Service agreement, discovery, invocation, composition, delivery, etc.
  – SaaS and service system integration
    ▪ In-house to SaaS (main app is in house)
    ▪ SaaS to in-house (SaaS is main app)
    ▪ SaaS with partial SaaS solution hosted on premises
    ▪ SaaS to SaaS integration
  – SaaS and enterprise on-premises systems integration is critical
    ▪ Not preferable to move all app/data to public accessible SaaS platform (cost, privacy, sensitive information, management)
    ▪ SaaS platform requires on-premises resources to be useful

SaaS - Major enterprise software trend in 2008 (rated #1 by Mckinsey & SandHill Group)
  – Salesforce.com
  – Microsoft (S+S, software + (Internet) service)
  – Amazon, …
The Trend of “as-a-Service”

Platforms/solutions are moving towards “as-a-Service”

- Software-as-a-Service (SaaS)
- Platform-as-a-Service (PaaS)
- Communication-as-a-Service (CaaS)
- Database-as-a-Service
- Monitoring-as-a-Service
- Network management-as-a-Service
- User Interface-as-a-Service
- Integration-as-a-Service
- Authentication-as-a-Service …

The common theme in this trend is: services and services computing, in addition to or to elevate from the more conventional low level constructs/methods
Web Services for Communication

- **Enable communication-as-a-Service**
  - Leapfrog to the paradigm of service-oriented communication
  - Make communication infrastructure consistent/submerged with the infrastructure of the Web
  - Open and extensible to support existing and emerging applications in unified communication

- **Challenges and issues**
  - Communication is stateful and relies on stateful context, e.g. session, to establish endpoints association and control
  - Communication is intrinsically two-way, full duplex interaction
  - Communication relies on some signaling protocols below the service description level, e.g. H.323, SIP, etc.
    - Is it possible to advance web services from the service level to the level communication signaling protocol?
  - Communication endpoints must be networked to be useful, and what is the communication networking infrastructure in communication as a service?
  - ...
WIP (Web Services Initiation Protocol)  
- Web Services based Communication over IP

- WIP (Web Services Initiation Protocol): Communication over IP based on a single stack of web services
- Meta-services/Base services based SOA architecture, utilizing WS-Session (ECMA-366/ISO 25487), WS-Addressing, WS-Eventing, etc.
- In WIP, media signaling is modeled by a special web service event subscription between event sink and event source
- Fully web services based communication protocol without dependency on non-web service methods for session establishment and signaling
- Extensible with web service protocols to achieve end-to-end security as web services endpoints
- Agnostic to the transport protocols, e.g. communication services can be enabled through HTTP, JMS, etc.

(Ref: W. Chou, Li Li, Feng Liu, ICWS’06, ICWS’08)
WIP for Communication Over IP

WIP A

wss:StartApplicationSession()

wss:StartApplicationSessionPositiveResponse(SessionID)

wss:Subscribe(SessionID, Offer)

wss:SubscribeResponse(SessionID, Answer)

wss:Subscribe(SessionID, Confirm)

wss:SubscribeResponse(SessionID, Ack)

Media Path

wss:StopApplicationSession(SessionID)

wss:StopApplicationSessionPositiveResponse()

A initiates call to B
A ends call
WIP for SaaS+CaaS
- A new communication and distributed computing paradigm
WIP Phones as Services in Enterprise

Business Applications

Web Service Registry (UDDI)
- publish
- link
- SOAP message (signaling)
- control

Web Search (Google)
- find
- Index (HTTP GET)

WIP endpoint
- find
- SOAP message (call control)

Media devices
- control
- media (RTP)
Web Convergence
expose web resources following REST approach

Web browser → Search engine → UDDI site

HTTP GET

HTTP Response:
hypermedia representation of resources

WIP endpoint (URL)

Root
- Registry
- Services
- Presence

WSDL File
UDDI

- support service discovery by business applications (businessEntity, businessService, binding, tModel)
- support wip address resolution for wip endpoints (wip:wuchou@research.avayalabs.com → http://123.4.5.6:8080/)
- serve as public key repository (wip address → public key) and on/off-line presence server
- wip endpoint self-registration (publication)
- publication by one message and find by two messages (drill-down)
- can support many-to-many wip address to service location (URL) mappings

model many-to-many associations of wip addresses and wip endpoint in UDDI:
\[ n1 \rightarrow u_1, u_2; n_2 \rightarrow u_2, n_3 \rightarrow u_1, u_2 \]
WIP for Communication Over IP

Figure 2: Diagram of WIP endpoint
Mobile Services-Computing Endpoint

**Critical Needs for Services Computing**

- Support advanced services computing applications on mobile endpoint
- Utilize mobile endpoint for distributed computing, as high-end mobile phones/handsets are getting pervasive with non-trivial computing power
- Support SaaS (software-as-a-service) and CaaS (communication-as-a-service)
- Enable new distributed services computing applications, especially SaaS.
WIPdroid (WIP+Android)
– A communication enabled mobile computing endpoint for SaaS and distributed computing

- **WIP (Web Services Initiation Protocol) endpoint**
  - Fully web services based service-oriented communication supports voice, video, multimedia/multimodal, presence aware, social network integration, etc.
  - Communication-as-a-service for seamless integration in SaaS applications

- **Android**
  - Google’s initiative of OHA (open handset alliance)
  - Application development environment with Google’s own Dalvik (java) virtual machine
  - User Interface, utilities, and interfacing to Google applications, e.g. maps, web browser, XMPP messaging, location, etc.

(Ref: Wu Chou, Li Li, “WIPdroid - …”, Proceedings of SCC’08, Vol. 2)
Google’s OS for mobile devices
WIPdroid Architecture
Red color indicates components developed by us

Android Emulator

- **Views:** login, call, presence, notification, advisory
- **Activities:** login, call, presence, advisory
- **Data:** contact, preference
- **Service:** request, event

Android SDK

- **XMPP service**
- **HTTPClient**

Platform OS

- **JMF**
  - WIP Switch (2SAP Endpoint Edition)
  - HTTP server
  - WS-Session
  - WS-Eventing
  - CSTA

Audio, video over RTP

SOAP over HTTP

HTML over HTTP (GET)

XML over TCP/IP

XMPP

HTTP request

XML over TCP/IP

HTTP request

XML over TCP/IP

HTTP request
2SAP (Two-way Web Services Application Proxy)  
– A middleware for web services enablement

Meta-Services/Base-Services based SOA Framework

☐ **Base-Services**
  - Provide specific functions in a particular domain
  - Extensible for various applications, e.g. CSTA Web Services (ECMA-348), Paraly X Web Services, etc.

☐ **Meta-Services**
  - Enforce rules for certain *common aspects* of base-services, e.g. Session (WS-Session), event (WS-Eventing, WS-BaseNotification), security (WS-Security), reliability, etc.
  - It can be synthesized with one or multiple base-services to enable a real application, but it has no real physical effect on its own.
2SAP (Two-way Web Services Application Proxy) – A compact web services enablement middleware

- **stateless vs. stateful**: disks indicate stateful resources shared by all services
- **core vs. extension**: ovals indicate extensions customizable by base services
2SAP Architecture: Events Handling

- Topic tree abstract event sources
- Event queues enable asynchrony and reliability
WIPdroid – GUI Interface

- Call and Call Control based on WIP (full web services based communication)
- WIPdroid mobile platform (two-way web services as well as mobile endpoint)
- GUI binding with Android phone interface
- Presence based on WIP presence engine that supports Parley X rich presence profile
WIPdroid – Presence Aware Communication

- Support rich presence of Parley rich presence profile
- Four presence categories including:
  - Activity
  - Privacy
  - Space
  - Sphere
- 14 presence states of Parlay
- Extensible and generic presence framework for applications in intelligent communication, social network, etc.
WIPdroid – Presence Setting GUI

- Presence setting GUI for presence state in Activity category
- Presentity can set its present status
- Watcher can subscribe and to receive the presence information of the Presentity
- Binding with Android phone GUI
WIPdroid – Presence Subscription GUI

- Watcher GUI to subscribe presence information of the presentity, e.g. Wu Chou
- Presence setting going through WIPdroid
- Communication to the Presentity through web services
- Binding with Android phone GUI
WIPdroid – Presence GUI of the watcher

- Watcher GUI of the presence information of the Presentity
- Presence information going through WIPdroid of the Presentity to the WIPdroid of the Watcher
- Communication between WIPdroids in two-way web services
- Binding with Android phone GUI
What is achieved by WIPdroid

- Mobile service computing for SaaS
  - Mobile SOA web services endpoint with real-time communication capabilities for SaaS and SOA platforms/applications
  - An extensible web services infrastructure, 2SAP Endpoint Edition (2~3MB), that enables mobile endpoint to become mobile SOA/SaaS endpoint, e.g. Android or other endpoints

- A new type of multi-mode phone
  - One from the binding with the physical mobile environment, e.g. GSM, CDMA, 3G, etc.
  - One fully web services enabled, service-oriented communication over IP mode from WIP.
  - New applications in SaaS, CEBA (communication enabled business applications), intelligent communication, etc.
Summary

- Cloud computing – Exciting research area and we are deep in the cloud!
- SaaS (Software-as-a-Service) – Top trend in enterprise software and we need to advance further
- Web services in communication – It is a start of a new paradigm of full web services based communication
- SaaS (mobile) endpoint for distributed services computing is an emerging area, and WIPdroid approach can be applied to other endpoints.

We are in an exciting time for web services and services computing.
Thank You