Web Services

An Historical Perspective

David M. Sherr

VP, Advanced Technology Group

Charles Schwab & Co., Inc.

charles SCHWAB

I wish to thank the people at Waters for the opportunity to present before my "home" crowd yet again.

Before I start, I wish to remember our colleagues and good friends who perished two years ago. By a stroke of luck, I was late to the conference that day and was in the lobby of the Millenium Hotel when the Apocalypse began. Thank you to my guardian angel. My daughter gave me an original water color of an angel watching over the Downtown skyline. Between it and a 1975 picture of the same skyline buildings reflecting into the East River, I have erected a shrine in my office. In July, when I visited Schwab facilities in Jersey City, I looked across the Hudson and realized that even though the Towers are gone, I still can see five buildings in which I worked, two of which I helped to build. So, Wall Street still physically and spiritually exists for me.

When asked to talk, I choose middleware because it is what I worked the hardest at to get understood in the 10 years I spent from 1987-96 that I spent at Citicorp Investment Bank, Shearson Lehman American Express and JP Morgan doing Capital Markets trading systems, architecture and R&D work.

One lesson learned is that even though we as technologists can "see" clearly how to apply technology to business problems, a combination of legacy systems and organizational impedance slows us down considerably.

Thus, I titled this talk.

Purpose of this Presentation

- **Examine the evolution towards Web Services over the past 25 years**
- **Counsel patience in implementation**
- **Invite commentary**

7/22/2008

2

Web Services as an idea has been the industry searching for standards-based interfaces to facilitate distributed processing across diverse systems.

The Story

What is a Web Service?	*The right level of abstraction is key*
Web Services Are Not New in Concept	*Thinking "outside the box"*
Architectural Evolution of the Enterprise Message Bus	*Coupling processes, activating remote functionality*
Where is the Industry and Schwab?	*Adoption rate depends on many factors*

charles SCHWAB

This is my Story and I'm sticking to it!

First, we need to establish just what a "web Service" is. Regardless of how you define it, it is important to abstract the notion to the right level. Technologies come and go quickly, but architectures hang around for a long time. Does *tn3270* ring a bell? Second, "Web Services" is not a new concept. It is a matter of thinking and operating outside the box—the processor box that is. I hope to convince you that the concept of "service" goes WAY BACK. Third, Architectural, Web Services derives from the idea of an enterprise message bus. This is all about how we couple (actually de-couple) processes to activate remote functionality. As a minor footnote, in the days of living completely within one box, viz., the IBM mainframe, the notion is not too far away from CICS function shipping, except it is asynchronous and not synchronous.

Last, we will end with looking at the current state of the Industry—in my humble opinion—and what we are doing at Charles Schwab & Co., Inc. Adoption rate of Web Services will depend on many factors, both specific to each enterprise as well as maturity of the vendor offerings.

What is a Web Service?

The right level of abstraction is key

charles SCHWAB

Let’s get to it without further ado.

Schwab's Definition of "Web Service"

- **An Interface Description:**
  - WSDL
- **A protocol**
  - SOAP
- **A transport**
  - http
- **Possibly, a Service Locator**
  - UDDI
    - *Dynamic/Static service binding is a separate issue*

7/22/2008 5

We could use an internal proprietary description language, but WSDL seems to be fine as the standard.

For maximum flexibility, the particular protocol and transport should be deployment parameters. This the realization of the essential abstraction.

Like its immediate predecessor incarnation, Corba, which we will look at in slightly more detail in a few minutes, location of the desired Service can be a matter of dynamic or static binding. What we learned is that we need to make dynamic binding easier.

Sun's Jini is an example of a proprietary architecture and supporting capabilities that achieve dynamic service location and binding.

From the Object Management Group, we have the Model Driven Architecture movement that is leading us this way. I eagerly await this next phase of Web Services definition and generation.

Web Services are Not New in Concept

Thinking "outside the box"

charles SCHWAB

While "services" began in one box, as distributed computing became a reality, we had to think of what to do outside of a specific processor box.

Thinking about Services, “Inside the Box”

- **When apps were totally contained inside one processor, we had resource managers rendered as**
  - Print Services
  - File Services
  - Security Services
  - Time Service
  - Database Services
  - Communication Services
- **Facilitated by post 1970 OS’s (MVS, Unix, VMS, WangOS, etc.)**
- **All totally proprietary, all your computing “eggs” in one vendor’s basket**
- **Connections amongst different systems required special hardware connectivity, except for ASCII/EBCDIC tapes or slow serial lines**

Thinking about Services, "Outside the Box"

- **As distributed computing matured into the '80s, these services needed to be coordinated and federated among many processors using different connectivity: "A Virtual Tower of Babel" with many of the same results.**
  - SNA
  - DECNet
  - WangNet
  - TCP/IP
- **But eventually TCP/IP prevailed as a standard by the end of '80s, leading to the commercialization of the Internet and the emergence of WWW by 1994**
- **By 1988 unix had gained commercial acceptance, thus building the Industry movement towards standards-based distributed computing**
- **Vendors then "exported" the idea of services as the model for distributing computing hastening the …**

charles SCHWAB

Evolution toward Web Services

- **Gen 0, 1980: Unix**
  - unix-unix execution
  - Connectivity
    - ethernet
    - Serial Port
    - IP
- **Gen 1, 1990: Distributed Computing Environment**
  - Interface Definition Language
  - Remote Procedure Call
  - Stub/Skeleton Compile time binding
  - IP connection

In the beginning There was ....

Gen 2, 1993: Common Object Request Broker Architecture

- Interface Definition Language
- Object Request Method Invocation
- Pre-Run Time Binding
- Dynamic binding existed, but was too arcane for widespread use

Gen 3, 2000: Web Services

- Dynamic Location and Binding: UDDI
- XML based description: SOAP
- IP based transport: http

And today we have ....

charles SCHWAB

This is the whole story in one slide: four generations of distributed computing architecture and technology and 20 years to get to where we are today.

Architectural Evolution of the Enterprise Message Bus

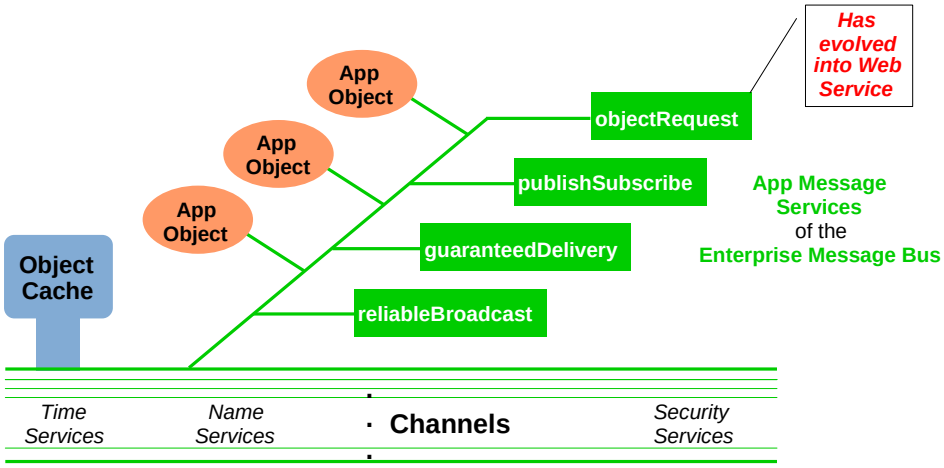
Coupling processes, activating remote functionality

charles SCHWAB

Here's the beef. A ten-year old idea looks thoroughly modern if the names are updated. Same problem, different day.

Enterprise Message Bus Architecture (circa 1993)

- Advanced n-tier Client/Server Messaging



Indulge me for a few moments while I travel down memory lane. In Oct 1993, I was invited to present a proposal of what a software major unix vendor should develop to help Lehman with its systems—we ought between $40-50 million a year of unix boxes at that time. I proposed that they construct middleware according to the architecture shown and called it the enterprise message bus. To me at the time, it was the logical extension to all the proprietary toolkits we were using. Create the EMB abstraction so we could swap out the proprietary technologies underneath. It was an amalgamation of DCE and Corba. The technology strategy was to separate infrastructure from application programs to simplify the application development process. (All of us still fight this battle today!). The App Message Services were fit for purpose capabilities:

•reliableBroadcast was for Quotes distribution

•guaranteedDelivery for order and trade placement

•publishSubscribe for information delivery

•objectRequest for intra/interprocess method dispatch.

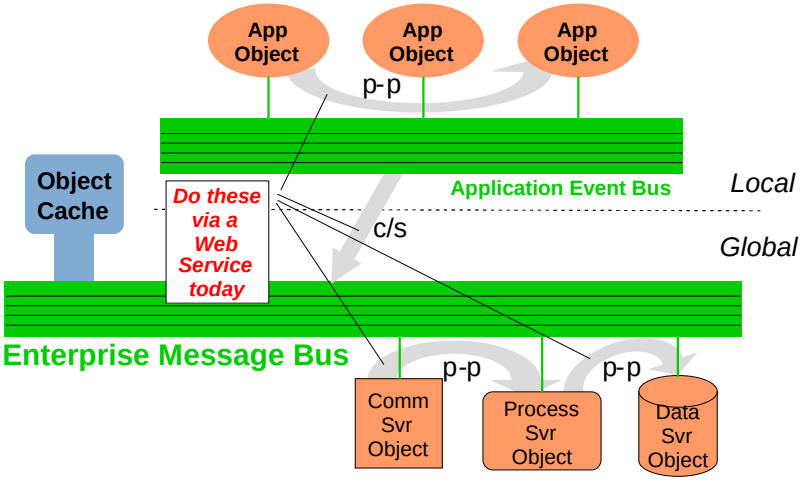
The *Time*, *Name* and *Security Services* were provided by the EMB which could support multiple channels. The Object Cache assisted in improving the performance of the systems built using the Architecture.

It is far to say, I believe, that objectRequest has evolved into Request/Reply which is realizable using Web Service technology today.

Turns out, they thanked me and went about doing what they did before: sell boxes and let third party vendors provide proprietary pieces of the EMB architecture which never worked well together.

Architecture for Collaborating Application Processes (circa 1993)

- Peer-Peer (p-p) and Client/Server (c/s) mechanism support



charles SCHWAB

A second part of the EMB Architecture involved handling local versus global message handling. The main point was to abstract local collaborating application components into event sharing via a special form of the EMB called an Application Event Bus where the components operated more or less in a peer-peer fashion with the global resource access was done in a client/server manner. The global resources also had the capability to interact in a peer-peer fashion as well using an EMD.

As noted, these message (event) sharing has evolved into Web Services today. Sigh! If I only had then, what is promised today!

Where is the Industry and Schwab?

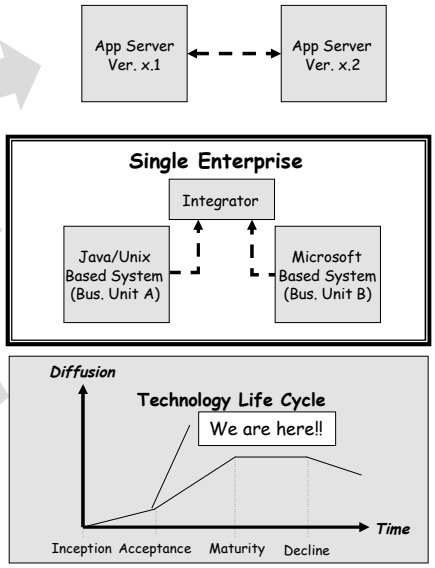
Adoption rate depends on many factors

charles SCHWAB

So where have we come to? What can we expect to do? How fast can we do it?

Where is the Industry?

- **Many are doing a small number of internal Web Services (SOAP/http/no UDDI) for very mundane and practical reasons, e.g.,**
  - Within one internal enterprise area, connect applications (EJBs) because the application server versions are incompatible
  - In B2B portal fashion across two internal enterprises, connect a Microsoft-based data center to a Java/Unix-based data center
  - *The few trans-Internet successes really demonstrate that as a technology, Web Services are at the end of Emerging, i.e, they are just reaching the turn up of wide scale adoption (in Moore's term "The Chasm" needs to be leapt successfully)*



charles SCHWAB

Despite the handful across-the-Web successful examples of Web Services, there are a few initial applications of Web Services that solve current, practical problems. That is, they deliver value but do not do it in the aspirational manner promised by the vendors of Web Service technologies. What this tells me as an advanced technologist is that Web Services are still an immature technology, barely out of the Inception Stage in its Life Cycle. What remains to be seen is how long the run-up from Acceptance to Maturity. According to Information Week, the VC's have made a major bet in 2001 funding to the tune of $974 million. This did not include what the major players invested themselves, probably in the billions by now. I ask, have we seen a positive return on these investments yet? Have Web Services created that much value in the consumer community? Where are the Research Analysts when you really need them?

Building an Enterprise Bus at Schwab

- **Introduced by the "3rd Wave Architecture" (circa 2000)**
  - A Service Oriented Architecture (SOA)
    - patent pending by Schwab
- **We have had several successful Proofs of Concept and Pilots with Web Services**
  - Within the Enterprise
  - Thus, the motivation to extend existing SOA with Web Services infrastructure at the logical level with implementation in a bus
- **The Bus includes the following mechanisms:**
  - Messaging (Publish/Subscribe): Asynchronous
    - An Architecture of Record and High Level Design completed in 2001
    - Implemented in 2002 for medium/high QoS
  - Request Reply Bus: Synchronous ***Web Service***
    - Platform Transparent System Access (PTSA) Architecture of Record completed in 2002
    - High Level Design, review complete, under development 2003

charles SCHWAB

Here is a brief description of the Schwab experience. We see the value, but are looking for the tangible benefits before we make a major investment. We are investing, but slowly to the point where we see the positive returns. As you can see, we have followed the historical evolution to this point. Learn as we go. Return as we go. Starting with the Service Oriented Architecture, followed by Asynchronous Messaging and finally to the Synchronous Web Service. One foot in front of the other. Leading, not bleeding edge.

What is the Current State of Web Services?

- **Web Services are more talk than walk**
  - More about how we agree to interact than how our systems actually interact
  - Current standards are far from set
  - W3C and WS-I political wars are not over!
- **Some App Server vendors are pushing heavy EJB containers for this lightweight purpose of invoking.**
  - Will new lightweight providers like Cape Clear and Mind Electric sneak in the door with less complex infrastructure?
  - .Net offers such a lightweight environment as well
- **Methods and form for orchestration and integration of Web Services need to be agreed upon**
  - A new Standards War brewing?

charles SCHWAB

Why are we cautious? Uncertainty is still in the specific form of the technology and in the vendors who will emerge as winners.

A Probable Roadmap:
Nine areas to address to achieve flexible, production-grade Web Services

1. Security
2. Performance
3. System Management
4. Service Level Agreements
5. Debugging and testing in a distributed environment
6. Platform choice: App Server based, Corba based, or *de novo* built
7. Functional Reuse
8. Transactional Assurance
9. Dependency management among the collaborating services

charles SCHWAB

What do we have to look forward to? This is the approximate order that the issue areas will probably be solved.

The first five are being solved today in vendor offerings; last four are yet to be addressed adequately by offerings.

This Roadmap is two years old already. It was articulated it in the September2002 RiskWaters magazine story where yours truly was one of many interviewed. If the last two years pace is any indicator, it should be another two-three years before we see Web Services as certifiably in the Acceptance Stage of the Technology Life Cycle. I do believe some have leapt Geoffrey Moore's Chasm. Stay tuned. Details at 11.

Thank you for you attention today.

Questions? Comments? Observations?