PLANNING ASSUMPTION

11/12/2002

Web Services Adoption Outlook Improves

Uttam Narsu and Phil Murphy

Giga Position

Recent changes in the Web Services standards landscape have improved the outlook for enterprise adoption of Web Services in the next three years. Enterprise adoption of Web Services will follow a modified Moore adoption cycle, with a bifurcation between external (inter-company) and internal (inter-company usage). External usage will generally follow internal usage, with the lack of standards for security, operational management, and transactions remaining as the chief barriers to widespread external usage. While vendors and standards address the external adoption barriers in the coming 12 to 24 months, internal adoption of Web Services moves forward, but only where business needs and potential gain outweigh the effort, cost and risk of early adoption.

Those caveats notwithstanding, progressive organizations have already begun analyzing their application portfolios to determine which business functions (new/existing) can be exposed as Web Services, and how services and legacy applications will coexist. Expect Web Services to gain momentum as an integration technology for internal applications in the 2003 to 2005 time frame, but wider adoption of Web Services for external, inter-company use will depend on many external factors (vertical XML standards, adoption by exchanges, performance improvements) and will lag behind by several years. Adoption by technologically conservative organizations will occur at the end of this decade. Most organizations will layer Web Services over existing systems; the full transition to service-based development and a service-oriented architecture will take the balance of this decade to play out.

Proof/Notes

Despite its promise of a simpler, more universal approach to inter-business application integration, the hype of Web Services currently outpaces reality. Outlandish predictions have emerged postulating that Web Services are revolutionary, that all application development will be based on Web Services in the near future and if you don't board the Web Services train now, you will be left at the proverbial station. This is clearly nonsense, and our user polls show that customers don't buy any of this hype (see IdeaByte, Web Services: Determine What Is Hype Before Deciding, Jost Hoppermann). Any such broad statements about Web Services adoption have to make a distinction between the use of Web Services internally within an organization or externally between partners, customers or suppliers. We discuss this in greater depth below.

From a pragmatic standpoint, simply building services isn't enough: A mindset of "If you build it, they will come" or even "If you build it, they will reuse it" is out of sync with today's tight economic times. Business partners communicating via newly constructed Web Services must not only buy in to the idea of Web Services, but must also adopt business model changes to support them. While Web Services can be used quite happily for simple application or data integration, one should never do so casually. To achieve a true services orientation will require that we rethink development and design processes to fit this new paradigm that is similar in concept to object-oriented (OO) and component based development (CBD), but at a much larger level of granularity. In today's economic climate, adopting the latest technology buzzword for emerging value is a difficult proposition at best. But with the hype properly counterbalanced, Web Services offer a compelling longer-term vision whereby business functions within and across enterprises interoperate without regard to their technological underpinnings. As such, they will, when mature, provide a major step forward for integration.

Recent changes in the Web Services landscape (see IdeaByte, Web Services Interoperability Group: Watch This Space, Uttam Narsu and other references for details), since an earlier version of this Planning Assumption was published have brightened the outlook for enterprise adoption of Web Services. The formation of the Web Services Interoperability Organization (WS-I); the re-emergence of the World Wide Web Consortium (W3C) as a player in Web Services standards creation; the creation of the WS-Security specification; the release of SAML 1.0; the continued and deepening vendor support of Web Services; the migration of UDDI activity to OASIS; the accommodation of external XML standards such as IFX to Web Services; and the creation of the WS-Transactions, WS-Coordination and BPEL4WS specs have all played their part. But the biggest event was the working draft release of the Web Services Basics profile by the WS-I in the last few weeks. The profile (see http://www.ws-i.org/Profiles/Basic/2002-10/BasicProfile-1.0-WGD.htm) offers users a clear, consistent definition of Web Services, and a base upon which to continue their pilots and prototypes. This release solidifies WS-I as the most significant Web Services standards body for end-users. Once finalized, WS-I's Basics profile will act to decrease the risk of adopting Web Services.

Web Services adoption will vary widely by industry and depth of adoption and whether it is intended for external or internal use. Financial services, automotive manufacturing and other supply chain-intensive industries will lead the way for externally deployed Web Services, while more mainstream industries, such as utilities and insurance, will lag well behind in the depth of their adoption (they may perhaps, employ Web Services only internally in their portal development efforts). Surprisingly, government is making strong commitments to Web Services at both the federal and the local level in the United States; expect similar trends in Europe and Canada. Across verticals, the most successful organizations will evaluate the potential benefits of Web Services according to their business needs and their ability/tendency to adopt new technology, and will set preparations in place for a more progressive and natural evolution to Web Services in the coming years.

The Seductive Vision

The vision of Web Services is that they will help IT organizations assimilate all the previous waves of technology (which still exist in many companies) into a services layer that will be broadly interoperable and platform-agnostic. Organizations will construct building block units of functionality (services) that will be assembled into Web applications by developers who have prior knowledge of the interface requirements. (With services like **Intuit**'s Quicken BillPay, there is precedent for connecting applications over the Internet using proprietary interfaces — what we might call "web services" with lowercase to distinguish them from the emerging Web Services standards based on SOAP, WSDL and UDDI. Web Services, by contrast, create the standards necessary to generate the network effect to drive broader adoption.) The business benefit of this approach is that future application development efforts are streamlined — since much of the application already exists in the form of services, developers simply glue the services together in a way that makes sense within the context of the new application.

The Difficult Case of External Web Services

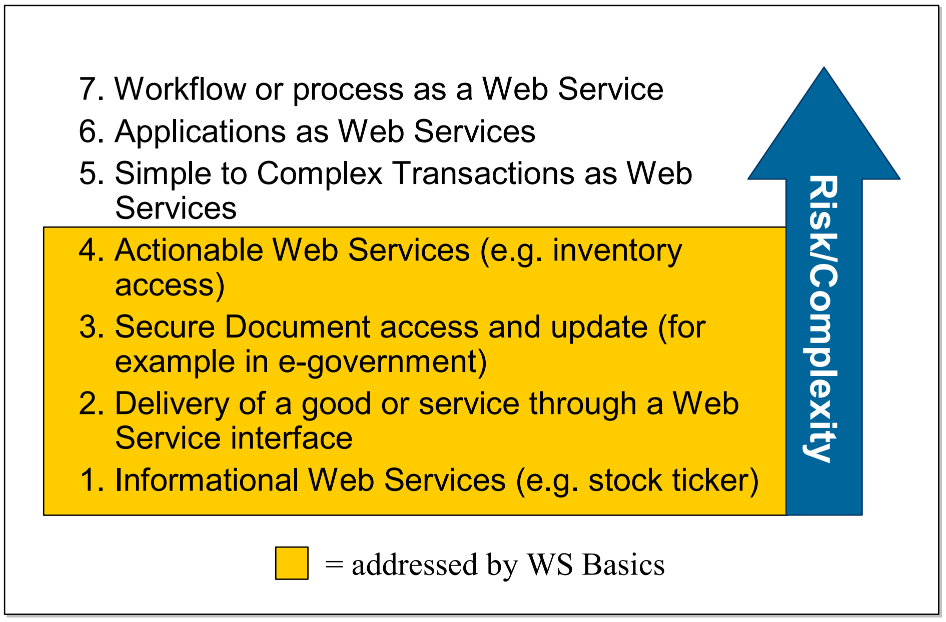
But achieving the aforementioned vision will be difficult. Much of the present hype concerns external Web Services — those services that are deployed and discovered over the Internet. The current Web Services standards are immature, and many are not very interoperable, which doesn't bode well for deploying external Web Services. More telling is the issue of security. Giga's surveys have consistently identified security as the No. 1 limitation of current Web Services initiatives (see IdeaByte, Viable Web Services Business Models Do Exist, but Require Mature Technology to Fly, Mike Gilpin). Security and the lack of reliable messaging over the Internet will likely be addressed in the coming 12 to 24 months. The emergence of SAML and the migration of WS-Security to OASIS ensure that a single set of Web Services security standards should emerge from OASIS in 2003. Although it's unlikely that all the additional standards in the WS-Security vision (WS-Policy, WS-Trust, WS-Privacy, WS-Authorization, WS-Federation and WS-Secure Conversation) will be finished in 2003, early work shows promise that the first quarter of 2004 will see those specs emerge. Until then, organizations will have to solve the issue of security much as they've solved it with their Web sites.

Yet, few organizations will run the risk of being first to expose financial and other "meaningful" transactions as services to the Internet, without the ability to enforce a service-level agreement. For example, when Thomson Financial deployed Web Services interfaces to some of their financial data, they did it through **Grand Central's** Web Services Network. According to a recent Giga survey, quality of service (QoS) runs a close second to security in the minds of those considering Web Services. QoS and security together help determine trust. Unless and until the requests for services and their respective responses can survive all manner of Internet routing glitches, network outages, machine failures and other impediments to successful and verifiable completion, public trust will remain an elusive target. Without robust and proven directory and security features that address authentication, authorization and audit requirements, there will be no public trust. Without public trust, widespread adoption will not occur, however compelling the technology.

The Case for Internal Web Services

With the above limitations, it is no wonder that organizations are cautious about where and when they first deploy Web Services. It seems reasonable to assume that adoption will follow the path taken by classic Web technology, along a continuum from low-value, non-mission-critical to high-value, mission-critical. Some examples of Web Services follow, from lower complexity and lower risk to higher complexity and higher risk (it's worth noting that these examples can potentially be either internally or externally deployed Web Services; for the case where it's external, the need to provide security raises the risk and complexity level over the internal form):

Figure 1: Web Services Examples Handled by the Basics Profile

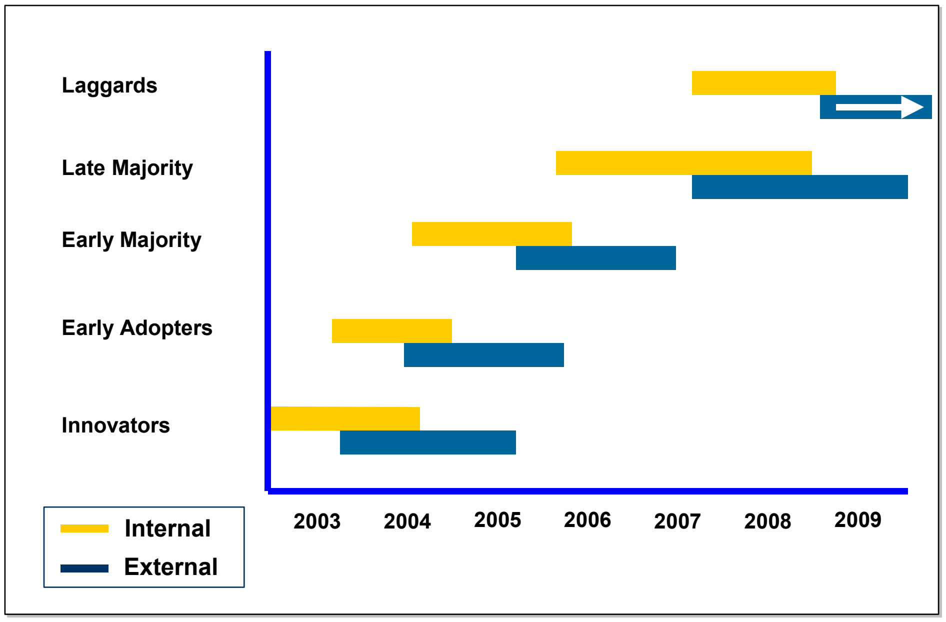

7. Workflow or process as a Web Service
6. Applications as Web Services
5. Simple to Complex Transactions as Web Services
4. Actionable Web Services (e.g. inventory access)
3. Secure Document access and update (for example in e-government)
2. Delivery of a good or service through a Web Service interface
1. Informational Web Services (e.g. stock ticker)
Risk/Complexity
= addressed by WS Basics

Source: Giga Information Group

Figure 1 depicts some examples of Web Services, from lower complexity and lower risk, to higher complexity and higher risk. The shaded box highlights the examples that can be addressed by the WS-I's Web Services Basics profile (with custom security, since security is not yet addressed by WS-I). It's worth noting that these examples can potentially be either internally or externally deployed Web Services; for the case where it's external, the need to provide security raises the risk and complexity level over the internal form. For some organizations, simple informational Web Services will be the first deployed, especially if the demands for security and demands on the underlying infrastructure are light. For these companies, this will take the form of replacing a previous generation of Web-based technology (server-side scripting, CGI and the like). Others will have doubts that there is business value in replacing a technology (server-side scripting) that works. For these organizations, creating access paths to stove-piped applications will be the compelling driver behind Web Services. Still others will be driven to adopt Web Services through adoption of a technology such as portal technology that is being transformed by Web Services. In other words, it is difficult to generalize about Web Services adoption patterns, since they will be largely driven by business need, vertical industry and the organization's adoption profile toward technology. Still, if we had to identify a dominant use of Web Services, we forecast that the most prevalent use of Web Services during the next five years will be as a standards-based approach to application integration — first internally, then with carefully chosen business partners. For the present, Web Services will not reduce the demand for integration servers since they clearly provide additional value in their support for mapping, routing, transformation and publish/subscribe; Web Services, however, will transform some aspects of the role these servers play (see Planning Assumption, Service Brokers: An Important Option for Implementing Service-Oriented Architectures, Mike Gilpin).

Given these points, Giga predicts that the "innovators" and "early adopters" of Web Services technology will develop and deploy Web Services for internal use in the 2002 to 2004 time frame. In 2004, the "early majority" join the fray, and in 2006 the "late majority," followed by the "laggards" several years hence. Because of the technology issues with security and messaging, external services follow internal service usage as depicted in Figure 2.

Figure 2: Projected Web Services Adoption Rates



Source: Giga Information Group

The factors/risks affecting the timing of this adoption model follow:

1. A double-dip recession, further delaying a return to normalcy of IT budgets
2. Performance problems with Web Services that prohibit or delay their use in certain situations
3. Continued standards-fragmentation, especially at the higher end of the Web Services stack, despite the efforts of the Web Services Interoperability Organization

4. Incomplete or delayed vendor support
5. The risk that your current product architecture will be incompatible with Web Services
6. The risk that your organization will be unable to absorb Web Services technology or adapt to its implementation (perhaps due to skills)

If any of these take place, then these dates will shift even further out for your organization.

Preparing for Web Services

Despite the fact that Web Services are a future technology concern for many organizations, progressive organizations have already begun preparing for them, if only to understand and articulate the value of Web Services (see Planning Assumption, The Business Value of Web Services: Brainstorm Now, Implement Later, Randy Heffner and Mike Gilpin). When to start preparing depends on the state of your IT environment and your time frame for implementation as indicated in Figure 2. Since business demand for functional interoperability should be the driver for Web Services projects, the most likely candidate applications are those that the business has been eyeing for some time with the hope of modernizing and integrating with newer applications. IT-driven demand for Web Services may also be valid to drive small proof-of-concept (POC) projects that flatten the learning curve for future applications development. Recognize that reuse should not be the primary goal for POC projects; instead, use these projects to make early mistakes and adjust processes and procedures to streamline future development efforts. In well-funded IT organizations, these POC applications can begin immediately, while other organizations will wait until the economic climate improves.

But simply adopting the Web Services standards doesn't build services. Sure, you could take existing application APIs or existing application data and expose them through the magic of Web Services to a wider audience. But you won't be doing *service-based development*. Service-based design identifies and rationalizes candidate services with business objectives. Not every function will be a service, nor should it be. In theory, a service boundary should align with commonly needed functions, represent a complete unit-of-work and have a clear and communicable end result. To be good candidates for reuse, the functions should be in fairly constant demand. Examples of simple services may be returning a customer balance, credit status or order status; each is valuable in many applications and returns a fairly discrete answer. Reporting functions can also be good early examples of services that provide business value, yet carry little risk.

With candidate services identified, examine the underlying technology of those applications: Are they stable and mature, is it possible to wrap the underlying application functionality as components so that services can be layered cleanly on top? How do the standards tap the existing systems — do the applications have a viable API, can they be accessed via the presentation layer using Web-to-host tools or is application mining a viable approach? How large a factor is transaction latency? Tactical approaches minimize organizational and process change in favor of quickly adopting Web Services for high-value business needs. A strategic approach will focus more on the transformation of the organization and process, to accommodate a services orientation, with each line of business analogous to a service bureau. These service bureaus identify their business "plug points" — where they naturally interact to exchange discrete information. The plug points and their relationships equate to service interfaces that will enable the business to reconfigure itself. Adopting a services orientation requires you to identify the cohesive business interfaces that constitute a service as well as the loosely coupled relationships between the service units. Successfully adopting a strategic approach will require a strong architectural focus (see IdeaByte, Web Services vs. Enterprise Architecture: Strategy Depends on the Level of Organizational Cooperation, Randy Heffner).

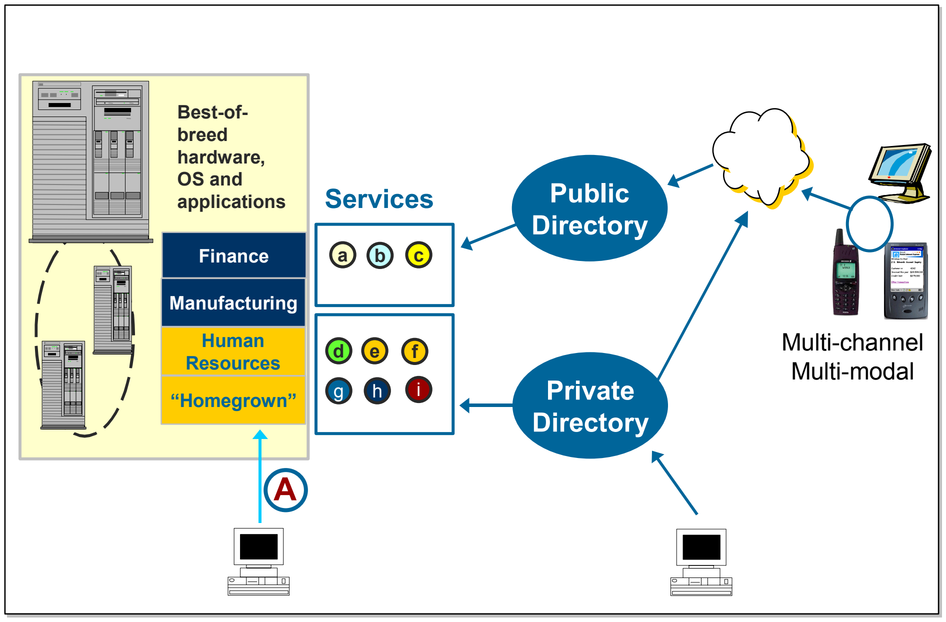
Adopt Web Services with a controlled and balanced process, including education, pilot groups and POC projects. Roll them out to wider audiences as driven by the success of early projects. In organizations that favor infiltration as opposed to more formal methods of technology adoption, attempting to enforce a more rigid process may create more problems than it attempts to solve. Use a process that fits your culture while leaning toward the most formal process your organization will tolerate. Recognizing that infiltration does in fact occur is half the battle. Infiltration can be managed through a try-learn-adapt-iterate approach.

Looking to the Future

Very often, it's useful to look to the future to gauge the likely impact of a technology "wave." A likely scenario for 2008 is depicted in Figure 3:

- Some discrete applications remain for operational access (see "A" in Figure 3).
- Multi-use functions that have value when exposed to new users are culled into services.
- Some are exposed externally.
- There is interplay between internal and external usage.

Figure 3: Web Services Impact — 2008



But we are a long way from that scenario now. In the interim, the *modus operandi* for all organizations, except innovators and early adopters of technology, is to operate with the following assumption: Acknowledge that Web Services are evolutionary, not revolutionary. As a technology, they will not replace everything before them, but will be used to make certain business functions interoperate in a more standardized fashion. Organizations with large legacy environments have both a great asset (a plethora of existing, high-value, well-tested business functions) and an albatross (a technology mix that needs to be streamlined and thinned). Given the people/process/technology nature of most IT challenges, these organizations will not only identify candidate services from existing services, but will also address Web Services knowledge/skills with a combination of training and proof-of-concept projects and mitigate some of the chaos by implementing process improvement around the new assembly/development paradigm. This approach positions organizations for the smoothest possible adoption of and evolution to better business interoperability by leveraging the power of a mature Web Services paradigm.

Alternative View

The next most likely scenario is that Web Services will be aggressively adopted but in what the Web Services Interoperability Organization characterizes as a “basic” profile. A basic profile means that stable versions of SOAP and WSDL are widely implemented by organizations for applications that are heavily stove-piped today and for which there is no other solution for revitalizing or rewriting these applications. For many users, this will equate to a “replace CORBA or DCOM, but don’t adopt service-based development wholesale” approach. We already see some signs that this approach is being pursued by very capital-constrained organizations. They do so because (1) Web Services technology is evolutionary, not revolutionary, and therefore easily understood by development staff, (2) the maturity of the current standards is sufficient and low risk for their adoption and (3) performance is sufficient for their needs.

Findings

Web Services hype doesn’t match reality — vendors and the trade press have blown the benefits and current-day capabilities completely out of proportion. The goal of self-describing, dynamically discovered and dynamically constructed Web Services, while a great technological vision, is at least four to eight years from being implemented, and due to “legacy customer behaviors,” may be a utopian scenario that we never fully realize. Web Services adoption patterns will be largely driven by business need, vertical industry and the adoption profile of the organization toward technology.

There will be no predictable path of Web Services adoption within organizations, though certainly organizations will likely proceed from simpler, lower-risk implementations to more complex and more high-risk implementations. We foresee internal Web Services usage emerging before widespread external usage. Once Web Services technology matures, it has the potential to revolutionize application integration methodology by providing a layer of abstraction between the technology that requests a service and the technology that provides the service.

Web Services technology currently lacks robust security, reliable messaging, QoS guarantees and support for sophisticated transaction models. The external form, which depends on UDDI, lacks directory services integration, meshed with sufficient security measures to warrant “public trust.”

Not every business function within every application will become a Web Service. While it may be possible to decompose every function within an application into a discrete service, that concept takes Web Services to a theoretical extreme given today’s technology.

Recommendations

Set the hype aside and evaluate Web Services technology for what it is today — a new approach to connecting applications that holds great promise, but contains certain aspects that remain immature for some current projects. Be wary of using the external form of Web Services and focus on whether, where and when your organization should adopt Web Services as an approach to internal application integration.

Begin planning for the adoption of Web Services by identifying high-value examples that fit the technology-adoption profile of your company. Innovators and early adopters of technology can act today; the trailing minority and majority should begin to consider pilot projects to prove/disprove the concept and its feasibility for adoption with their planning horizon. Focus on the WS-I’s Web Services Basics profile as the base upon which you build first-generation Web Services.

The security, QoS and reliable messaging issues make widespread exposure of meaningful transactions to the Internet impractical at this time. Focus on how Web Services can simplify the integration problems internally until these issues are resolved — not likely before the fourth quarter of 2003 at the earliest. Begin analysis of the WS-Security proposals to see what will be usable within your planning horizon.

Educate the organization's business leaders on the current and future capabilities of Web Services as outlined in Planning Assumption, The Business Value of Web Services: Brainstorm Now, Implement Later, Mike Gilpin and Randy Heffner. Let educated business-need drive which services get the attention of Web Services developers. Draw candidates for "services" from the highest-demand integration projects. Applications that were candidates for Web-to-host tools — the existing functionality is valuable enough to keep, but requires browser or multi-channel, multi-modal access — are likely to be excellent candidates for Web Services. In applications with no discernable API, Web-to-host tools provide a number of options to expose business functions to the Web Services standards, and in that context, enable Web Services to operate in conjunction with custom-built legacy applications.

References

Related Giga Research

Planning Assumptions

The Business Value of Web Services: Brainstorm Now, Implement Later, Randy Heffner and Mike Gilpin

Web Services: Emerging as a Broad Internet Application Integration Standard, Bigger Than Just Microsoft. NET, Randy Heffner, Mike Gilpin, et al

Service Brokers: An Important Option for Implementing Service-Oriented Architectures, Mike Gilpin

A Realistic Assessment of Web Services, Mike Gilpin and Ken Vollmer

IdeaBytes

IT Trends 2003: Web Services Standards, Uttam Narsu

Informational Web Services Are Useless Without Semantic Agreement, Uttam Narsu

Viable Web Services Business Models Do Exist, but Require Mature Technology to Fly, Mike Gilpin

Web Services: Determine What Is Hype Before Deciding, Jost Hoppermann

Web Services vs. Enterprise Architecture: Strategy Depends on the Level of Organizational Cooperation, Randy Heffner

Web Services Interoperability Group: Watch This Space, Uttam Narsu

WS-Security Sealed as the Foundation for Secure Web Services, Randy Heffner

IBM and Microsoft Take Command of Enterprise Web Services Standards, John Rymer

Relevant Links and Other Sources

Web Services Interoperability Organization (www.ws-i.org)

Glossary

API — Application programming interface

B2C — Business-to-consumer

BPEL4WS — Business Process Execution Language For Web Services (www-106.ibm.com/developerworks/library/ws-bpel/)

CGI — Common Gateway Interface

CORBA — Common Object Request Broker Architecture (www.omg.org/)

DCOM -- Distributed Component Object Model (www.microsoft.com/com/tech/DCOM.asp)

J2EE	Java 2 Enterprise Edition
SOAP	Simple Object Access Protocol (see XMLP)
QoS	Quality of Service
SAML	Security Access Markup Language (www.oasis-open.org/committees/security/)
SSI	Server-side includes
UDDI	Universal Description, Discovery and Integration (www.uddi.org)
WSDL	Web Services Description Language (www.w3.org/TR/wsdl)
WS-Coordination	Web Services Coordination (dev2dev.bea.com/techtrack/ws-coordination.jsp)
WS-Transactions	Web Services Transactions (msdn.microsoft.com/library/en-us/dnglobspec/html/ws-transaction.asp)
WS-I	Web Services Interoperability Organization (www.ws-i.org/)
XMLP	XML Protocol (www.w3.org/2000/xp)
XML	Extensible Markup Language