



The Mainstream

An article from the IBM @server zSeries software newsletter

Find the current version of The Mainstream here ibm.com/software/zseries/mainstream

Web-savvy integration of the On Demand model

*from The Mainstream, Issue 10 - 2004
The IBM @server zSeries and S/390 software newsletter*

Companies can do business faster and cheaper when leveraging Information Technology (IT) to automate their business processes. These IT systems each offer discrete services or functionality of particular business value; however, in order for other internal or external systems to have access to these services, the respective systems first need to be integrated. When two companies form an agreement and integrate their respective systems to obtain business value, this is usually known as Business-to-Business Integration (B2Bi). In this kind of relationship, Company A would become the integration partner of Company B.

From a historical technical perspective, successful integration projects have delivered solutions in spite of, and perhaps with the assistance of, a diverse set of protocols and technologies. This has generally made implementation complex and costly because of all the customization required. It has also made it impossible to build one system interface for your application that all your future integration partners can use. The goal of the integration architect is therefore to achieve the most efficient means of leveraging these existing IT systems to either create or reuse existing system interfaces when integrating with other system interfaces.

The role of integration within On Demand

An On Demand business is intended to be able to respond with flexibility and speed to any customer demand, market opportunity, and external threat by being responsive, variable, focused, and resilient. It achieves this by implementing systems with the following characteristics:

- *Integrated*
Systems are seamlessly linked across the enterprise and across its entire range of customers, partners, and suppliers.



ibm.com/software/zseries/mainstream

- *Open*
Different systems can work together and link with devices and applications across organizational and geographic boundaries.
- *Virtualized*
Makes the best use of technology resources and minimizes complexity for users, such as using grids to make the collective power of the computing resources in the grid available to anyone in the grid who needs them.
- *Autonomic*
It can respond automatically and work around problems, security threats, and system failures in order to be self-healing.

The rest of this article focuses on the integrated and open system characteristics. For more information on the On Demand model (see Resources).

On Demand integration and Web services

In order to allow the integration of business applications to facilitate the On Demand provisioning of services, you need a set of industry standards upon which to build your technology. Within the last few years, Web services have emerged to provide a set of standards that allow disparate applications running on a variety of platforms and technologies to seamlessly communicate. In other words, Web services provide the integrated and open characteristics required for On Demand, due to open standards compliancy. The On Demand model requires that Web services are leveraged as the integration and invocation interface to your application, which, coupled with a Service Broker, provides an open standards implementation of a Service-Oriented Architecture (SOA). In essence, your application is now able to be invoked using industry standards, which greatly reduces the complexity in integration and increases its reusability.

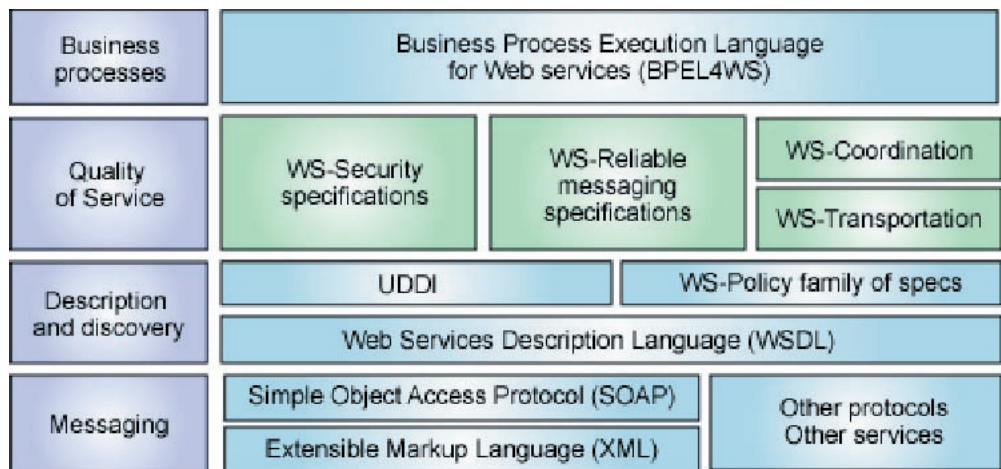
The term Web services has been quite loosely defined until quite recently when the Web services Interoperability Organization was established (see Resources). It created and published the Web Service Basic Profile 1.0, which defines the composition of a Web service and how to interact with it. This is an important achievement as adherence to the Basic Profile reduces integration

ibm.com/software/zseries/mainstream

complexity between Web services, especially when using tooling to produce and consume Web services-related artifacts.

Figure 1 gives you a high-level view of the existing Web services technology stack. It should help you identify which standards are applicable for a particular concern (such as Messaging). For more information on these standards, check out the Web Services Standards section on the developer Works Web site (see Resources).

Figure 1: The Web services technology stack



This is obviously good news to those embarking on building integration applications, but what about those of us who are already vested with pre-existing assets and solutions? IBM has made great strides in providing tools and technologies that enable you to wrap a SOA interface around existing assets. IBM® WebSphere® Studio Application Developer V5.1 bundles the IBM WebSphere SDK for Web services which provides wizards that can turn your existing business logic into Web services. In addition, the Web services Invocation Framework (WSIF) allows for Web services to be invoked over a variety of protocols, and this capability will be further enhanced by the future release of the Enterprise Service Bus. This is encouraging in that you can leverage existing WebSphere MQ infrastructure implementations, which provide guaranteed delivery, while SOAP over HTTP (the most common Web service implementation) does not.



ibm.com/software/zseries/mainstream

Service Orchestration specifies which services are invoked in particular scenarios and in what order (much like a workflow). You could, for example, create a Purchase Order Business Process Orchestration which involves first invoking several of your own internal services and then several of a trading partner's external services. Once all the responses have been obtained, collated and coordinated, the final step of the orchestration could return the finished result to the originating service. This choreography ensures that each step in the transaction is executed according to the policy. The Web Service Business Process Execution Language (BPEL or BPEL4WS) provides a standard with which you can script the orchestration and choreography of your Web services in XML markup.

Enable On Demand integration

Now that you understand that Web services can provide an open standards implementation of a reusable system interface that is targeted at seamless integration, take a look at the 4-step process required to enable the On Demand integration-related aspects of your application:

1. *Expose services internally*
2. *Expose services externally*
3. *Integrate internal services*
4. *Integrate external services*

In order to actually put the 4-step process into practice, you first need to start internally (within your own company and applications). Decide which of your internal applications need to be exposed as Web services so that they can be either leveraged internally or be used later when you begin external integration. For instance, is there a legacy Payroll application implemented using proprietary technologies and protocols that prevent your other more modern applications from making use of it? Once you create a Web service interface for the Payroll application, it will be exposed to any internal Web service application that could benefit from using it.

Once you've cleaned up your own house, it's time to look externally. What about that Business-to-Business (B2B) interface you've always wanted to put in place to help you save money when working with your partners? By Web services-enabling your partner interface, you can provide your Web Service Description Language (WSDL) file describing your interface to your partners, so

ibm.com/software/zseries/mainstream

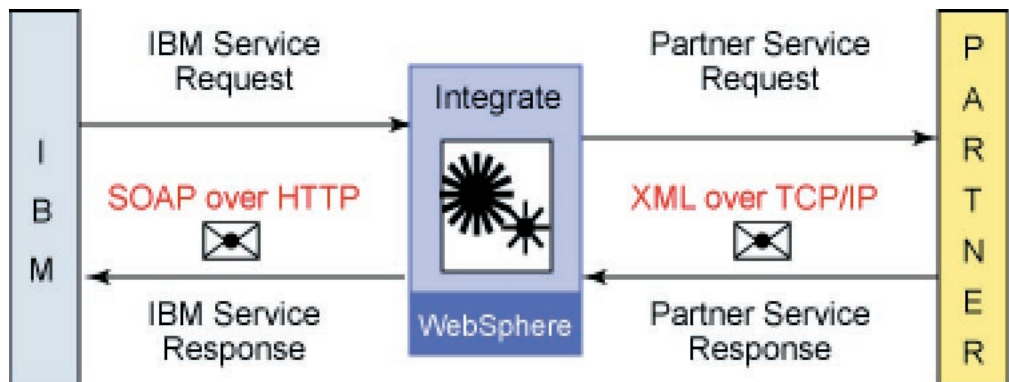
that they can begin integration with your external partner interface. In addition, you can choose to publish your WSDL on a public Universal Description, Discovery and Integration (UDDI) Registry (see Resources for a link). This registry is similar in concept to the Yellow Pages, where interested parties can look you up and obtain your WSDL description to discover how to integrate with you.

What problems do you face?

Before you start looking at your existing products and solutions, take a brief look at the problems you might potentially encounter.

Perhaps not all of your partners' interfaces are Web services-compliant. Earlier in the article it was mentioned that the term Web services was, for awhile, loosely defined. A lot of your partners might have integration interfaces such as XML over TCP/IP, XML over WMQ, or XML over FTP. In addition, they might operate with a proprietary XML message protocol (in other words, not SOAP) and their own proprietary XML Message Schema. Since all partners might have their own unique XML Message schema within their SOAP document, you have to handle the translation (by applying mapping rules) between your proprietary message schema and theirs. Some message schemas are built using attribute-based XML which is particularly difficult to map compared to non-attribute-based XML, and some message protocols use RPC-based Web services while others use Document-Literal.

Figure 2: Issues involved with integration





ibm.com/software/zseries/mainstream

In addition, some partner interfaces are designed to communicate synchronously, some are designed to be purely asynchronous, and some have complex transaction lifecycles that are a mixture of both. When asynchrony is used, often partners will embed their own correlation logic within their message protocol in order to coordinate which response correlates to which request. This requires you to further customize your interface in order to comply, and getting this right can be complex. WS-ReliableMessaging, WS-Coordination, and WS-Transaction are planned industry specifications to address standardizing the transactions in these solutions (see Resources).

With the advent of WS-I Basic Profile more developers are becoming truly compliant; however, partners with maverick interfaces do still exist and you will need to cater to them when building your external integration partner interface.

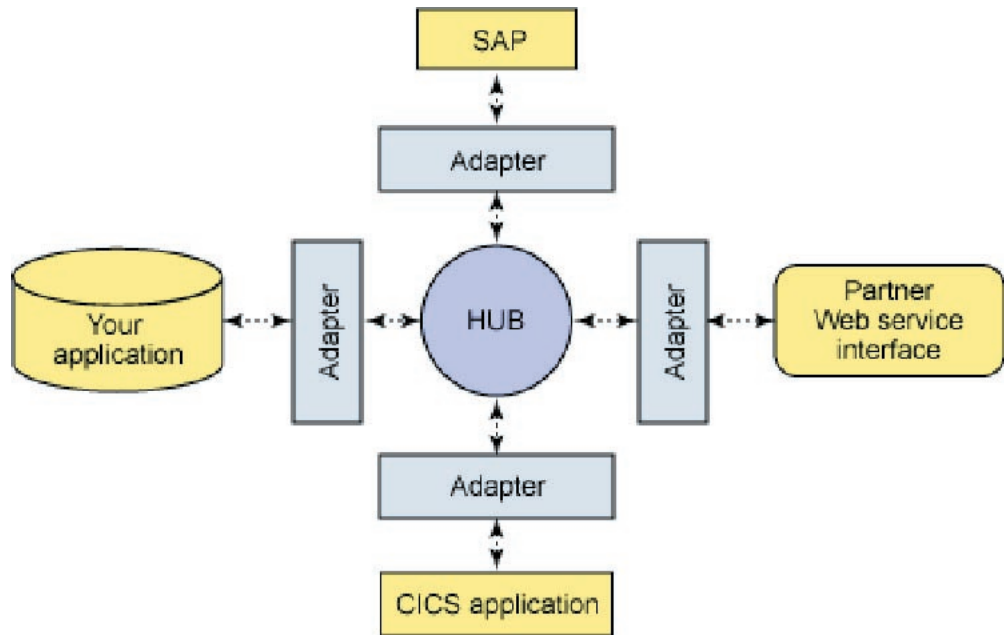
Available solutions

A Hub and Spoke architecture is best suited to building an integration interface. This is a recognized design pattern that specifies the use of flexible adapters to handle communication with maverick interfaces and protocols (the spokes). Additionally, a Hub is needed to orchestrate the process flows and interaction between your various internal and external services. IBM has a number of products available that meet this requirement. In fact, because of a spate of recent acquisitions, there are so many products and technologies that provide Web services integration and orchestration that architects will most likely be able to find a tool that works best for their environment.

The WebSphere Process Choreographer is available in IBM WebSphere Studio Application Developer Integration Edition V5.01 (Application Developer) and WebSphere Application Server Enterprise. The Process Choreographer uses Flow Definition Markup Language (FDML) to script orchestrations and provides a plug-in that allows you to import and export the FDML as BPEL.

ibm.com/software/zseries/mainstream

Figure 3: The Hub and Spoke architecture



The WebSphere Business Integration Suite is typically used in the Enterprise Application Integration (EAI) space and bundles the WebSphere InterChange Server V4.2.2. The WebSphere InterChange Server used to be known as IBM CrossWorlds and is a powerful Integration and Orchestration Hub. In addition, it interoperates with the WBI Adapters (the spokes), which facilitate communication between a wide range of protocols ranging from WebSphere MQ to Web Services. InterChange Server provides Collaborations, which is the InterChange Server term for Orchestration, and also provides limited BPEL import and export functionality.

IBM has done a lot of work to streamline the On Demand product offerings and the next evolution of the tooling is available within Application Developer Integration Edition V5.1, which allows you to use the WebSphere Process Choreographer to create BPEL-based orchestrations. IBM has also just released WebSphere Business Integration Server Foundation which now provides the strategic server runtime for executing BPEL orchestrations created in Application Developer. To learn more about WebSphere Business Integration Server Foundation (see Resources).

ibm.com/software/zseries/mainstream

Conclusion

In this article you learned how integration has historically been used within IT Systems, how the role of integration fits within the On Demand model, and how to implement On Demand integration using Web services. If you'd like to learn more about the On Demand model spend some time going through the IBM On Demand Web site. IBM developerWorks also has some excellent articles targeted at Integration-related technology in their SOA and Web services zone (see Resources). And if you'd like to dig deeper and see whats coming next, take a look at Rick Robinson's articles on the Enterprise Service Bus (also in Resources).

Web Resources

- [Start out at the IBM On Demand Business Center to learn more about On Demand.](#)
- [Browse the Web services Interoperability Organization \(WS-I\) Web site.](#)
- [Check out the developerWorks Web Services Standards section.](#)
- [Publish your WSDL on a public Universal Description, Discovery and Integration \(UDDI\) Registry.](#)
- [Standardize the transactions in your solutions by referring to the WS-ReliableMessaging, WS-Coordination, and WS-Transaction \(WS-AtomicTransactions and WS-BusinessActivity\) specifications.](#)
- [Get all the latest information about IBM's strategic Integration Middleware platform at the IBM WebSphere Business Integration Server Foundation Web site.](#)
- [Dig deeper and see whats coming next—take a look at Rick Robinson's articles on the Enterprise Service Bus. \(developerWorks, June 2004\)](#)
- [Access Web services knowledge, tools, and skills with Speed-start Web services, which offers the latest Java-based software development tools and middleware from IBM \(trial editions\), plus online tutorials and articles, and an online technical forum.](#)
- [Want more? The developerWorks SOA and Web services zone hosts hundreds of informative articles and introductory, intermediate, and advanced tutorials on how to develop Web services applications.](#)



ibm.com/software/zseries/mainstream

About the author

Stephen Watt is a Software Architect working on middleware integration solutions with the IBM Network Service Providers. He works within the e-Technology Center for IBM Global Services at the lab in Austin, Texas. Steve works extensively with Web Services and the latest WebSphere products and spends his time figuring out how emerging products and technologies can improve existing integration solutions within Service Delivery. Prior to working for IBM, Steve spent several years consulting in the Middle East and working for startup companies in the USA and his native South Africa. Steve is the co-author of *Professional.NET Programming for J2EE Developers* and *The ASP Maintenance Handbook* (Wrox Press) and has also published several articles at *ASP Today*. You can contact Steve at swatt@us.ibm.com.

Copyright © 2004 IBM Corporation.

IBM, the IBM logo, the On Demand logo and WebSphere are trademarks or registered trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.

ON DEMAND BUSINESS™