IBM White Paper
April 2006

Infrastructure Considerations for Service Oriented Architecture

Contents

2 Introduction
3 SOA Primer
3 Service Oriented Architecture (SOA)
5 Benefits of SOA
6 Web services and SOA
8 Enterprise Service Bus (ESB) and SOA
9 SOA and IBM's on demand operating environment
12 Infrastructure required to support a Service Oriented Architecture
12 SOA and infrastructure requirements
14 Infrastructure considerations for SOA
15 Infrastructure architecture
16 Integration
17 Enterprise systems management
19 IT security
21 Performance
22 Availability
24 Virtualization
27 Summary
27 IBM Infrastructure Services for SOA
29 Conclusion

Introduction

If you work in the IT industry, you've probably noticed an increased focus on Service Oriented Architecture (SOA) in the marketplace. Chances are that your business may also be doing something with SOA like a pilot or proof-of-concept. However, SOA is not a new concept, but a different approach to an old IT problem: how to design and build systems that are flexible and adaptable to the ever changing business environment? So what's different about it? Why is it hot right now? Thanks to matured standards, SOA is poised to fulfill promises that other application architecture approaches have only aspired to, but never delivered. It also reflects a concerted effort to have a comprehensive, end-to-end architecture that works within and between enterprises in order to respond to increasing business demands.

As businesses evolve to become more flexible and responsive (what IBM refers to as an "on demand business"), they have found that SOA and Web Services are key in enabling this transition. But they have also found that it is not enough to focus on their applications in this transformation – they also must ensure that their IT infrastructure can support the new architecture.

If recent IT history has taught us anything, it's taught us not to forget about the infrastructure. During their initial foray into e-business, many companies focused on ensuring that their Web sites had the "sizzle" to attract customers, but failed to take into account the new dynamics of the environment. As a result, their lack of attention to the infrastructure impacted the performance, security, and operation of their sites, and in the end, tarnished their brand image and cost them in lost customer revenue.

As businesses adopt SOA and Web Services, these new, simplified, virtualized and distributed application frameworks pose new infrastructure challenges that must be addressed. To ensure that the new applications can meet their performance, availability, scalability, security and management requirements, the IT infrastructure needs to be assessed and transformed.

This white paper outlines the infrastructure considerations related to implementing and operating a Service Oriented Architecture (SOA) application. It will interest any IT professional who is planning on implementing an SOA application in their IT environment. It will be of

Highlights

particular interest to those who are responsible for IT infrastructure or data centres such as Directors of IT Operations, IT Architects responsible for setting technology directions, or IT Infrastructure Specialists.

Before we outline the infrastructure implications for SOA, let's begin with a brief SOA primer.

SOA Primer

Service Oriented Architecture (SOA)

A Service Oriented Architecture (SOA) is an abstract application framework based on a set of design principles that takes everyday business applications and breaks them down into individual business functions and processes, called services.

These services are inter-related through well-defined interfaces and contracts between the functional units of the application. The interface is defined in a neutral manner that should be independent of the hardware platform, the operating system, and the programming language in which the service is implemented. This allows services, built on a variety of such systems, to interact with each other in a uniform and universal manner.

Service Oriented Architecture (SOA), using loosely-coupled systems, can help facilitate an agile and flexible IT system that can readily adapt to changing business needs and demands.

This feature of having a neutral interface definition that is not strongly tied to a particular implementation is known as loose coupling between services. The benefit of a loosely-coupled system lies in its agility and ability to survive evolutionary changes in the structure and implementation of the internals of each service that make up the whole application. Tight-coupling, on the other hand, means that the interfaces between the different components of an application are tightly interrelated in function and form, thus making them brittle when any form of change is required to the application.

The need for loosely-coupled systems has arisen from the need for more agile business applications. This is based upon business needs to adapt to changing environments (such as changing policies, business strengths, business focus, partnerships, industry standing, and other business-related factors) that influence the very nature of the business. We call a business that can act flexibly in relation to its environment an on demand business, where changes occur as necessary, on demand.

Highlights

Service oriented architectures are not new, but an alternative model to the more traditionally tightly-coupled, object-oriented models that have emerged in the past decades. While SOA-based systems do include individual services that can be built with object-oriented designs, the overall design is service-oriented. Since it allows for objects within the system, SOA is object-based, but it is not, as a whole, object-oriented. The difference lies in the interfaces themselves. A classic example of a proto-SOA system that has been around for a while is the Common Object Request Broker Architecture (CORBA), which defines similar concepts to SOA.

However, the SOA of today is different in that it relies on a more recent advance based upon the eXtensible Markup Language (XML). By describing interfaces in an XML-based language called Web Services Definition Language (WSDL), services have moved to a more dynamic and flexible interface system than the older Interface Definition Language (IDL) found in CORBA. Another difference between CORBA and SOA with Web Services is the much simpler application programming interface (API). Web Services are based on well known, open standards based technologies including the Hypertext Transfer Protocol (HTTP) and the Simple Object Access Protocol (SOAP). This is in stark contrast to previous technologies that were complex and more difficult to support.

Defining workflows and transformation points are key considerations in designing and implementing a successful SOA.

Web services aren't the only way to implement an SOA. As just explained earlier, CORBA is one other option. Message-Oriented Middleware systems such as the IBM® MQSeries® represent another option. But to become an architecture model, you need more than just a service description. You need to define how the overall application performs its workflow between services. More so, you need to find the transformation point between the operations of the business versus the operations of the software used in the business. Thus, an SOA should be able to relate the commercial processes of your business to your technical processes, and map the workflow relationships between the two. For example, the act of paying a supplier is a business process, while updating your parts database to include the newly supplied shipment is a technical procedure. Thus, workflow also plays a significant role in the design of an SOA.

Furthermore, the workflow of a dynamic business can include operations not just between departments, but even with other external partners, over which you may have no control. Thus, to be effective, you need to define the policies of how relationships between services should transpire, often in the form of service-level agreements and operational policies.

Highlights

Finally, all this has to work in an environment of trust and reliability to carry out the processes as expected according to agreed terms. So, security, trust, and reliable messaging should play a significant role in any SOA.

Benefits of SOA

An SOA lets you build, deploy and integrate these services independent of applications and the computing platforms on which they run, making business processes more flexible. This business flexibility can help bring you faster growth, lower total cost of ownership, and better access to timely and accurate information. SOA can help deliver additional asset re-use, easier management, and faster development and deployment. Because change is the only thing you can count on in today's business world, the agility to respond with speed to customer demands, market opportunities and external threats is more important than ever.

In addition to providing a flexible IT solution, a SOA can provide an important alignment and linkage between business and IT systems.

In addition, by raising the level of abstraction in application development from the object to the business service, it is now possible for those closest to the business to model, reuse, or recompose existing services to create value for a corporation. With modeling technology that can break down directly into code, there is a great opportunity to let those that best understand business play a greater role than ever before in the creation of business applications.

Businesses of every type are recognizing the value of an application that is componentized, interoperable, modular, and scalable:

- Componentized: Leverages standardized services interfaces for applications and resources
- Interoperable: Enables easier information exchange between applications and/or resources
- Modular: Mixes and matches, adds or removes, business processes and infrastructure
- Scalable: Starts with what you have and adds additional resources as needed

The benefits of an SOA are also realized by both business and IT as outlined below.

Highlights

Business Benefits	IT Benefits
• Business flexibility provided by increased granularity of processes enabled through services • Ability to more quickly create business processes and composite applications to respond to changes in the marketplace • Improved customer service using services with reduced worry about the underlying IT infrastructure	• Becoming a more responsive IT organization with a more secure and managed integration environment • Decreasing development and deployment cycle times through the use of pre-built, reusable services building blocks • Reducing complexity and maintenance costs with common services • Enhancing existing IT systems rather than replacing them

SOA can make it less expensive for many companies to link its business processes to those of your suppliers, vendors and other business partners. It also helps make it possible for individual customers and employees to buy or interact online with merchants and a host of other services, whatever their IT platform.

And if that weren't enough, SOA provides extraordinary flexibility by treating elements of business processes as components to be reused and combined in different ways to address changing needs and priorities. As a result, SOA helps make companies more agile and responsive to change.

Web services and SOA

Web services is a technology that allows applications to communicate with each other in a platform and programming language-independent manner. A Web service is a software interface that describes a collection of operations that can be accessed over the network through standardized XML messaging. It uses protocols based on the XML language to describe an operation to execute, or data to exchange, with another Web service. A group of Web services interacting together in this manner defines a particular Web service application in an SOA.

Web Services, which uses XML based on open standards, can offer flexibility, adaptability and integration of software applications across heterogeneous systems.

The software industry is finally coming to terms with the fact that integrating software applications across multiple operating systems, programming languages, and hardware platforms is not something that can be solved by any one particular proprietary environment. Traditionally, the problem has been one of tight-coupling, where one application that calls another is tied strongly

Highlights

to it by the function call it makes and the parameters it requests. In most systems before Web services, this is a fixed interface with little flexibility or adaptability to changing environments or needs.

Web services uses XML based on open standards that can describe any and all data in a truly platform-independent manner for exchange across systems, thus moving towards loosely-coupled applications. Furthermore, Web services can function on a more abstract level that can reevaluate, modify or handle data types dynamically, on demand. So, on a technical level, Web services can handle data much more easily and allow software to communicate more freely.

On a higher conceptual level, we can look at Web services as units of work, each handling a specific functional task. One step above this, these tasks can be combined into business-oriented tasks to handle particular business operational processes. This in turn allows non-technical people to think of applications that can handle business issues together in a workflow of Web services applications. Thus, once the Web services are designed and built by technical people, business process architects can aggregate them to solve business level problems. To borrow a car engine analogy, a business process architect can think of putting together a whole car engine with the car frame, body, transmission, and other systems, rather than look at the many pieces within each engine. Furthermore, the dynamic platform means that the engine can work together with the transmission or parts from other car manufacturers.

What rises from this last aspect is that Web services are helping to bridge the gap between business people and technologists in an organization. Web services make it easier for business people to understand technical operations. Business people can describe events and activities and technologists can associate them with appropriate services.

Web services can also facilitate reusability of applications and software, thus helping to optimize software use as well as ROI.

With universally defined interfaces and well designed tasks, it also becomes easier to reuse these tasks and thus, the applications they represent. Reusability of application software means a better return on investment and on software, because more can be produced from the same resources. It allows business people to consider using an existing application in a new way or offering it to a partner in a new way, thus potentially increasing the business transactions between partners.

Therefore, the primary issues that Web services address are the issues of data and application integration, and those of transforming technical functions into

Highlights

business-oriented computing tasks. These two facets can allow your business to communicate on a process or application level with your partners, while leaving dynamic room to adapt to new situations or work with different partners on demand.

SOA itself is an abstract concept of how software should be put together. It relies on the more concrete ideas and technologies implemented in XML and Web services to exist in software form. In addition, it also requires the support of security, policy management, reliable messaging, and accounting systems in order to work effectively. You can improve on it even further with the addition of distributed transactional processing, and distributed software state management.

The distinction between SOA services and Web services lies in their respective designs. The SOA concept does not exactly define specifically how services interact, just how services can understand each other and how they can interact. It is the difference between defining a strategy of how a given process is to be executed, and the tactics of how this is actually accomplished. Web services, on the other hand, have specific guidelines on how messaging between services needs to interact; that is, it is the tactical implementation of an SOA model most commonly seen in SOAP messages delivered over HTTP. Thus, Web services are essentially just one of many ways in which an SOA can be implemented.

While SOA can be implemented using other protocols and techniques, the optimal way to implement SOA is with Web Services.

SOA is not restricted to Web services, although we feel it is the best way to accomplish this. Other protocols that also directly implement service interfaces with WSDL and communicate with XML messages can also be involved in SOA. As indicated elsewhere, CORBA and IBM's MQSeries can now also participate in an SOA, using new features that work with WSDL. If two services need to exchange data, they will still need to use the same messaging protocol, but the data interfaces allow the same exchanges of information.

Enterprise Service Bus (ESB) and SOA

To establish proper control of all such messaging as well as to apply the needs of security, policy, reliability and accounting, there is a new software object that enters the picture of an SOA. This is the Enterprise Service Bus (ESB), which is responsible for the proper control, flow and optional translation of all messages between services, using any number of possible messaging protocols. The ESB is not absolutely required, but it is a vital component of properly managing your business processes in SOA. The ESB itself can be a single engine or even

Highlights

The Enterprise Service Bus (ESB) is an integral component that complements and helps manage business processes in a SOA.

a distributed system consisting of many peer and subpeer ESBs, all working together to keep the SOA system operational. Conceptually, it has evolved from the store-and-forward mechanism found in earlier computer science concepts such as the Message Queue, and distributed transactional computing.

An ESB can help optimize the delivery of information and services to help improve cycle time, reduce costs and improve IT flexibility. Based on open standards and industry leading technology such as IBM middleware, an ESB pattern provides interoperability between different platforms, programming models and software architectures. It aids in protecting and enhancing existing investments in applications, services and skills. The consistency of an ESB pattern enables incremental enhancement of your connectivity infrastructure, helping to reduce up-front investments and drive down maintenance costs. As your business needs change and integration opportunities arise, an ESB approach can help make your infrastructure more flexible, consistent and manageable.

In general the ESB is considered an infrastructure component, as it does not host or execute business logic. This is in contrast to components such as service requesters, service providers, and business service choreography whose role is to handle business logic.

The ESB is not a product but is rather an architectural pattern that offers a comprehensive, flexible and consistent approach to integration that is complementary to SOA. The ESB pattern is founded on and unifies message oriented, event driven and service oriented approaches to integration. It is an integral part of a Service Oriented Architecture and an on demand business.

SOA and IBM's on demand operating environment

An on demand business, one that is able to quickly respond to changes and is integrated across the enterprise and with key partners, needs an IT environment that can keep up with the pace. The on demand operating environment is IBM's blueprint for a new kind of IT infrastructure—one that can give your business the flexibility and power it needs to succeed. This guide will give you a quick overview of what it is, how it works, and how it can help your company operate on demand.

The on demand operating environment doesn't come in a box. It's not a product, a platform or a brand. It's a set of capabilities—hardware, software, technical

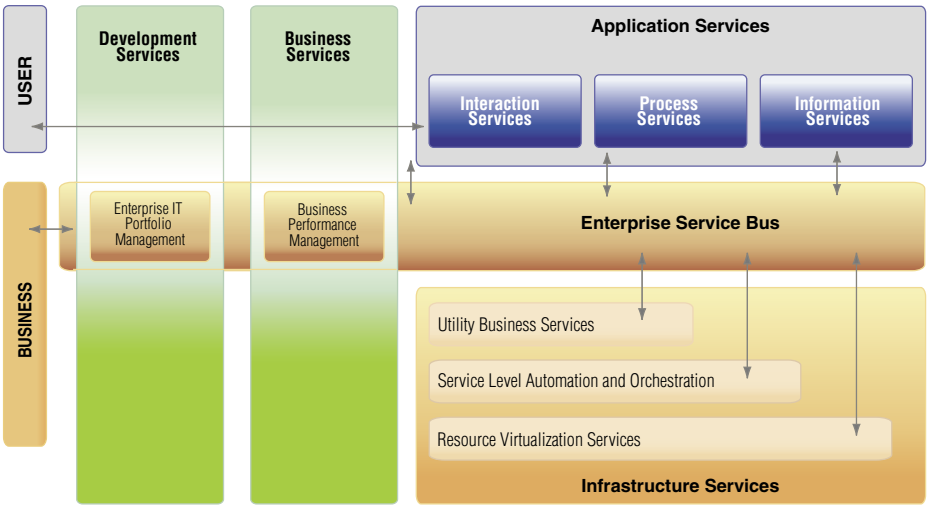
Highlights

expertise and more—that help companies build infrastructures that are more tightly integrated and more easily manageable.

What makes this computing model different? The on demand operating environment is modular rather than monolithic, connected by open standards rather than by proprietary ones. This means that your business can mix solution vendors together – including IBM and non-IBM solutions that adhere to these open standards. This allows your company to knock down silos, connecting disparate systems into a seamless, flexible infrastructure that can help optimize computing resources across the enterprise. It can improve traditionally low IT utilization numbers, unlock additional value in legacy systems, and allow new capabilities to be integrated into the environment more cheaply and easily.

IBM's on demand operating environment is based on SOA and helps provide flexibility to mix applications, processes and components.

Based around a Service Oriented Architecture, the on demand operating environment allows applications, processes and components to be mixed, matched or eliminated as needed. This flexibility helps to give organizations the power to quickly seize new opportunities and the resiliency to handle unexpected obstacles. The IBM on demand operating environment is SOA realized.



IBM's On Demand Operating Environment

Highlights

Each element of the architecture can help implement your environment capabilities and includes:

- **Development services** to enable an IT project team to collaborate in order to design, construct, integrate, test and deploy applications into the operating environment
- **Business services** that a business unit elects to project at its boundary, interfacing with its clients and partners
- **Application services** to integrate people, processes and data in order to service the business (also often referred to as integration services)
- **Enterprise service bus to** provide the infrastructure enabling mediated interactions between different service types in the environment
- **Infrastructure services** to provide the ability to define resources and management functions of the system in a standard and interoperable way, provision and operate those resources to defined service levels, and account for their usage in a utility model.

By using a modular approach, IBM's on demand operating environment can allow a business to evolve and adopt new IT capabilities at its own pace.

These new capabilities are highly modular, designed to be implemented gradually as your company evolves its technology to operate on demand. Their value, however, can be quite immediate. In the short term, they can help you speed ROI, cut costs and find new sales opportunities. In the long run, they can help transform your corporate IT infrastructures into powerful instruments to drive your business objectives.

The on demand operating environment based on SOA doesn't use technology for technology's sake, but as a framework to help build the flexibility, responsiveness and efficiency that today's business world requires. Using this model, a company can build an infrastructure that often lowers total cost of ownership, while maximizing the value of existing IT investments. It can give organizations the ability to evolve their systems without having to "rip and replace" hardware or software. This can save money and help to minimize disruption, so companies can more easily change their infrastructure, as quickly and as often as their business needs dictate.

But where do you start? Let's take a deeper look at the infrastructure requirements for SOA.

Highlights

Infrastructure required to support a Service Oriented Architecture

Solid infrastructure design and operations are the fundamental techniques for building infrastructures and managing Web Services in support of SOA. No matter what the application, infrastructure design always starts with an understanding of the non-functional requirements.

SOA and infrastructure requirements

Non-functional requirements for a business system address those aspects of the system that, while not directly affecting the functionality of the system as seen by the users, can have a profound effect on how that business system is accepted by both the users and the people responsible for supporting that system. Non-functional aspects of a business system cover a broad range of themes, which can include:

- Performance (including capacity)
- Scalability
- Availability (including recoverability and reliability)
- Maintainability (including flexibility and portability)
- Security
- Manageability
- Environmental (including safety)

Both functional and non-functional requirements must be considered in designing an effective SOA application.

Together with the functional requirements, these themes define the baseline against which the business system must be designed. In many cases, however, non-functional requirements receive minimal attention compared to the functional requirements. While it is true that the delivery of the required function is a fundamental requirement of a business system, without suitable consideration to the non-functional requirements, the SOA application may not you're your business needs.

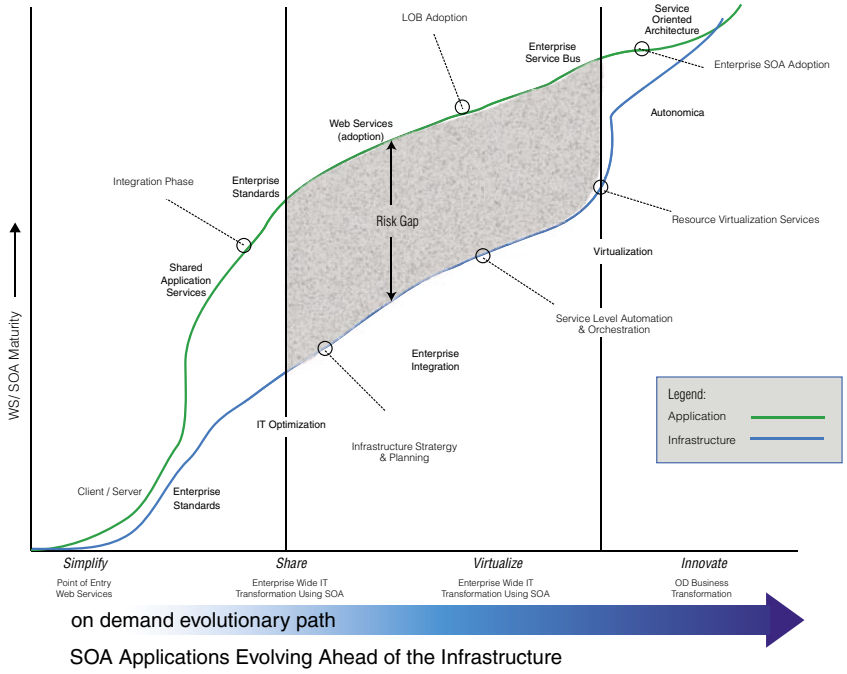
Fortunately, infrastructure services to support the business remain the same regardless of the application framework. Any applications that support Web services still require the basic services that an infrastructure provides. In terms of SOA, these infrastructure requirements include a focus on security, performance, management, availability, connectivity, and infrastructure design.

Highlights

As your business adopts a simpler application framework using SOA, you also need to transform your infrastructure to support SOA and Web services. This infrastructure does not change with Web services, but rather it evolves to support Web services. A majority of the infrastructure solutions used today can usually be used to successfully build, secure and manage Web services. The WS-standards, security techniques, application communication and management help define the differences in an infrastructure that is ready to support SOA. So the real question is when?

To help reduce the risk to the SOA application and thus to the business, make sure the IT infrastructure evolves to meet these new needs.

As the applications begin to exploit more mature Web services functions, the infrastructure transformation needs to proceed in parallel to complement the application requirements. If the infrastructure does not keep pace with the SOA applications, a risk gap will emerge leading to shortcomings in the infrastructure in its ability to support, operate, and manage the application.



In order to avoid creating this gap, infrastructures must evolve at the same rate to support the SOA applications and their requirements for production operation. In order to address this, one must look at the deeper infrastructure considerations for SOA.

Highlights

To support SOA, IT infrastructure capabilities need to evolve from being systems-focused to being services-focused.

Infrastructure considerations for SOA

The techniques used to build infrastructures today need to evolve to build an SOA and manage Web services. This infrastructure transformation occurs along the following seven domains.

Current Domains	SOA Specific Domains	Description
Infrastructure architecture	SOA Infrastructure Design and Enablement	Efficiently designs and delivers a secure, resilient and variable infrastructure
Integration	Enterprise Integration for Web Services and SOA	Bridges the gap between the application and infrastructure requirements
Enterprise systems management	Web Services Management	Manages & monitors Web services and ensuring their meeting service requirements
IT security	Infrastructure and WS-Security Solutions	Builds total solutions using best practices of infrastructure and web services security
Systems performance	Performance Testing and Management for SOA	Helps ensure that Web services meet performance requirements for SOA
Systems availability	Availability Management for SOA and Web Services	Helps ensure that Web services are available
Systems virtualization	Virtualization of Systems and Service Delivery	Delivers computing resources in a flexible, dynamic manner to support WS

If you see a common theme emerging from these considerations, it's that the infrastructure capabilities are evolved from being systems-focused to being services-focused. This is a fundamental shift of how IT views and delivers functionality – from "I manage a server, I manage a SAN, I manage a network" to "I manage IT services and components that support this business process". Let's look at each of these infrastructure considerations for SOA in greater detail.

Highlights

Infrastructure architecture

An infrastructure design for SOA and Web services is still structured around the business, applications and non-functional requirements. So in principal, you could tailor an existing infrastructure design to support new Web services requirements. Infrastructure for Web services is about providing the management, security, performance, availability, scalability and operations to support SOA and Web services. While these considerations are applied today to traditional systems design, they are applied here for SOA using a view towards services as well as systems considerations.

Fortunately, the design methods and reference architectures that infrastructure architects use today are still applicable to building a robust and mature infrastructure design for Web services and SOA.

This places an increased importance on the infrastructure architect obtaining the application requirements from the application architect. In the past, infrastructure architects have been able to avoid having a deep understanding of how applications function while simply focusing on system requirements. Knowledge about the application architecture, function and even the business process it supports are key to interpreting infrastructure design and operations impact. This is due to the fact that the application is more virtualized and distributed and may also involve elements provided by business partners. Having this awareness will in turn better position the infrastructure architect to design a solution that will meet the SOA non-functional requirements.

There are some key questions for infrastructure architects to consider in designing a solution that meets SOA's non-functional requirements for security, performance, management and availability.

Some key questions that an infrastructure architect should be asking are the following:

- Infrastructure design: How is this application going to be supported within my current infrastructure? How can the design of the infrastructure ensure that all requirements and expectations are met?
- Application architecture: What are the application requirements and how will this affect my infrastructure?
- Infrastructure and Web services security: How do we secure Web services and how does this affect my overall infrastructure security goals?
- Systems performance: How will Web services affect the performance for my applications? Are we doing XML transformation between our legacy and distributed systems?

Highlights

- High-availability: What happens if one of my services is unavailable? Where are my applications located and what are my dependencies between my business partners?
- Infrastructure and application management: How do I manage my Web services to tell if they're available & performing? How do I validate my applications and ensure they're meeting business goals?

These questions will begin to provide answers for the infrastructure requirements in support of SOA. Each of these question areas are discussed in greater detail in the following sections.

Integration

One of the goals of implementing an SOA is to provide the capability of integrating a set of loosely-coupled, dynamically deployed components implemented as services to meet rapidly changing business demands. But how do you integrate these services that are on multiple disparate technologies or perhaps even outside the walls of your organization for a business partner?

As mentioned earlier, at the core of SOA is the Enterprise Service Bus (ESB). This provides the flexibility of integrating and interconnecting a wide variety of services to be able to share messages, data, and transactions within and across the architecture. Communication services, connectivity services, and mediation services are all provided through the ESB.

An Enterprise Service Bus (ESB) can provide a foundation for integrating, interconnecting and managing services in your SOA environment in a consistent and flexible manner.

SOA also contains a set of services that are oriented toward the integration of people, processes, and information to provide the business flexibility to access functions from multiple, different end user types and devices, and access to information and data from multiple disparate sources, without the need to have direct access and connection to that data:

- **Interaction services** provide the capabilities required to deliver IT functions and data to end users, meeting the end-user's specific usage preferences, typically implemented utilizing portal technologies.
- **Process services** provide the control services required to manage the flow and interactions of multiple services typically through the use of business process choreography technologies.
- **Information services** provide the capabilities required to federate, replicate, and transform data sources that may be implemented in a variety of ways.

Highlights

This services oriented approach allows for many of the services to be provided through existing applications; others are provided in newly implemented components; and others are provided through external connections to third party systems such as through business partners. Existing enterprise applications and enterprise data are accessible from the ESB through a set of access services that provide the bridging capabilities between legacy applications, pre-packaged applications, enterprise data stores and the ESB.

The ESB capabilities are organized into three layers.

1. First, we have a bus connections layer that includes a rich set of client APIs, standard protocols, and adaptors. This layer enables the universal connectivity by ensuring that different services may connect to the ESB, whether they are Java, Microsoft®.NET® or others.
2. Second, we have a communication layer that provides messaging capability to carry messages between services. It helps provide high performance, scalable, available, and more secure messaging for all involved services independent of how they connect to the bus. It provides the flexibility of a range of messaging models as well as a variety of qualities of service.
3. Third, we have a mediation layer that provides the semantic glue between disparate services. As mentioned above, we expect services to connect in a number of ways and to be integrated to connect to each other. The mediation layer provides the flexibility to handle data format translation and other transformations necessary to enable communication between services such as XML transformation.

In addition to these three layers, the bus offers comprehensive management of services spanning all three layers. Establishing an ESB early on in your adoption of SOA is critical to laying the proper foundation and framework for integration within and amongst the services.

Enterprise systems management

Insight into the services layer of the composite application is required to support SOA as an extension of your existing Enterprise Systems Management (ESM) framework.

One factor in the ultimate success of SOA lies in the area of services infrastructure management. You will want to ensure that your enterprise systems management frameworks are extended to include the new SOA applications. Without the ability to effectively manage and monitor these applications, you will not have the necessary insight into your production environments to support SOA. Fortunately, SOA and Web services management is an extension of Enterprise Systems Management (ESM), which most

Highlights

businesses have deployed within their data centres. SOA management hooks into your existing ESM framework to help provide the insight, monitoring and management required to support these new application structures.

Treating services as manageable resources means applying the entire set of known systems management disciplines, including discovery, monitoring, version and change management, provisioning, policy management, security, service level agreements, and workload management to services, not just to the applications the services are combined to form or the IT systems whose resources on which they depend.

SOAs are frequently built using Web services, a platform-independent collection of protocols and standards that allows different applications or systems to exchange information. Because Web services are designed for reuse, they help improve development efficiencies and accelerate deployment. To provide effective management, Web services should be incorporated into the end-to-end management domain that supports the composite applications and the SOA infrastructure.

Web services management provides the administration, management, audit and reporting of the deployed Web services in the SOA. This can include:

- Visualization of Web services through a dashboard/management portal
- Analysis of historical Web services data (content and context)
- Visualization of ESB and messaging infrastructure
- Providing views for managing Web services, including their inventory and performance
- Providing Web service monitoring views
- Start/stop, audit and logging of Web services
- Managing Web services using thresholds

There are various tools in the marketplace to address the needs of SOA and Web services management.

There are various tools in the marketplace to address management of SOA and Web services. The key for most businesses is to understand their operational requirements and ensure that these tools provide the capabilities and functionality needed to meet their service level objectives. Whether you choose to use tools from vendors that specialize in this field such as AmberPoint, a company that produces a set of "solutions that make Web services production ready" or whether you choose products from end-to-end systems management software providers like IBM's Tivoli®, most usually provide open interfaces to link into your enterprise systems management facilities.

Highlights

For example, IBM Tivoli Composite Application Manager (ITCAM) for SOA can monitor, manage and control the Web services layer of IT architectures while drilling down to the application or resource layer to identify the source of bottlenecks or failures and to pinpoint services that take the most time or use the most resources. ITCAM for SOA is a core component of the IBM SOA Foundation Management Essentials, an integrated and open set of software, best practices, patterns and skills resources to get you started with service-oriented architectures.

IT security

The most common inhibitor to adoption of SOA and Web services is the lack of secure access to Web services and the lack of secure transmission of data between a client and a Web service.

The security environment is still disjointedly hardwired into organizational silos segmented into network security, perimeter security, desktop security, server security and application security. Point solutions solve a partial need but they don't work in unison. Hence, they can't appreciably lower system risk, improve platform integrity, or mitigate the risk of broadening access. Thus, the lack of integrated security management becomes a significant inhibitor for SOA adoption.

SOA adoption introduces new and unforeseen challenges with security integration, identity and security management. These can include:

- Multiple application platforms (WebSphere®, Microsoft or SAP)
- Multiple security domains (internal, external, business unit silos, extranet)
- Multiple security credentials (Kerberos, SAML, WS-Security, RACF)
- Multiple protocols (SOAP, HTTP/S, JMS, MQ)
- Lack of "thread of identity" across the services context

SOA security means spanning traditional silo security management systems and integrating them to ensure continuity across the system.

SOA applications must deal with the challenges of independent security and identity silos. The security solution needs to secure end user interactions as well service interactions (application to application). Security management needs to provide unified customer views for the composite application. The "thread" of user identity needs to be preserved end to end for auditing and compliance purposes.

Highlights

The challenges of security integration across application platforms, business services and infrastructure require that new forms of security and identity services be enable SOA applications to leverage "security as services." The "security services layer" needs to be an integral part of service management. These layers include the following:

1. **Authentication services** deliver identity and authentication services for both passive clients (browser-based) as well as active or rich clients such as desktops, portals and business integration components. Secure access to Web services resources has been largely based on transport level security (to provide confidentiality) using transport security methods such as Secure Sockets Layer/Transport Layer Security (SSL/TLS). For SOA, this must be extended to include the functionality in transport level security by allowing for authentication and authorization of requests based on transport security methods - for example, using mutually authenticated SSL to both build a confidential transport layer and to authenticate the requestor.
2. **Identity federated services** is a technology for brokering identities between companies or business units. Federated identity management is the set of business agreements, technical agreements and policy agreements that enable companies to partner to lower their overall identity management costs and improve user experience. It leverages the concept of a portable identity - the idea that your identity is not bound to a specific credential - to simplify the administration of users in a federated business relationship. Federation simplifies integration because there is a common way to share identities between companies and manage user sessions. Identity federation services within an SOA ensure that users have simplified access and single sign on to the composite application environment.
3. **Session management services** are key to providing consistency. As SOA transactions originate across various channels and protocols, it is important to have a common session management service that enables various SOA components to have a "common view" of the current user session. Such data can be used for single sign on, single sign off, auditing and reporting and to enable the services to implement policies such as inactivity timeouts, three-strikes-and-out, and other security policies consistently across various access channels.
4. **Authorization services** ensure that SOA components can apply consistent authorization policies for Web/HTTP/Java resources, Web services, SOAP (WSDL resources), MQ (Queues and Queue Managers) and even core infrastructure platforms such as UNIX and Linux servers. Authorization services in an SOA ensure that a common authorization abstraction model enables application platforms such as WebSphere, MS .NET, BEA and SAP to apply fine-grained authorization for these resource types.

Highlights

5. **Auditing services** provide common auditing services which ensure that security and change management activity across the infrastructure and SOA platform can be instrumented, collected, archived and reported for compliance against policies and various regulatory frameworks.
6. **Security token services** provide translation services for security claims. SOA applications transcend application platforms that may use different types of security tokens for expressing security claims. These tokens could be binary tokens or XML tokens and they vary between platforms. This service does the claims translation between various SOA components such as XML Firewalls, Enterprise Service Bus, Web services platforms (WebSphere, Microsoft .NET, SAP NetWeaver) and Business Integrator.
7. **Policy services** provides a centralized security policy service for centrally defining and managing security policies across HTTP, SOAP (WSDL), MQ and custom resources. This helps ensure that Web services security policies are enforced in a regular and consistent manner across multiple file types, application providers, devices and protocols. This facilitates the secure deployment of web services, allowing you to deploy your Web services-based applications more quickly and securely in heterogeneous environments.

Performance

By now you should be familiar with the primary benefit of SOA being business agility, but what about its performance? Performance should never come as an afterthought – it should be engineered into the solution. If not properly considered by both the application developers and infrastructure designers, your SOA performance may likely generate inefficiencies.

Combining performance engineering and performance management can help ensure that your SOA application will meet business service level expectations.

Since Web services SOA is mostly XML based, and XML is text heavy and verbose, consideration should be paid to addressing the challenges of increased metadata. Impact to the network, parsing requirements, security steps (which we've covered) and text translation must be considered when engineering a SOA solution. Beyond application design and optimization, various techniques exist to the infrastructure designer to deploy specialized tools such as accelerators or hardware-based network appliances such as IBM's DataPower SOA appliances to help address specific performance bottlenecks. However, software-based solutions may also provide performance enhancements and flexibility to customize and upgrade that cannot be provided by hardware solutions.

In addition to performance engineering, performance management functionality must also be considered. Today's business processes often depend on composite applications that span Web servers, J2EE application servers, integration middleware and mainframe systems. Although most

Highlights

businesses have traditional monitoring tools to manage individual resources at a high level, many lack an integrated solution to automatically monitor, analyze and resolve end-user response time problems. As a result, operations and development may take a long time to identify, isolate and fix transaction problems impacting customer satisfaction.

Tools such as the IBM Tivoli Composite Application Manager (ITCAM) for Response Time Tracking can help you avoid critical Web services performance problems by discovering, isolating and decomposing the Web services calls into the underlying components (Enterprise Java Beans, Servlets, Java Connector Architecture, DB2 etc.), allowing for true root-cause analysis of Web services performance and availability failures.

This tool also allows you to follow the path of a user transaction end-to-end across your business infrastructure. You can drill down each step that the transaction takes as it travels across multiple systems, and measure how each component of a transaction contributes to the overall response time. No matter what management tool you choose to use, insight into performance of a SOA application is important in order to be able to identify, analyze and correct performance issues.

Availability

Availability management becomes even more critical in SOA given the distributed nature and loose coupling of the application systems.

Your IT infrastructure is literally the heart of your business operations, so ensuring the reliability and availability of IT resources is vital for business success. Given the increasing complexity of applications and supporting infrastructures, intelligent performance and availability management tools are essential for proactive identification and resolution of IT problems before they impact your business results. Given the distributed nature of SOA and the dependencies on business partners that may also provide services, any type of Single Point of Failure (SPOF) can disrupt a SOA application's availability.

When implementing SOA, it becomes even more critical to understand the various aspects of availability and how they are impacted by the architecture. Service Level Agreements take on new meaning in an SOA, where it is possible that the service consumer is inside your organization and the service provider is outside, accessed via a Web service interface. The question is: How do you measure and manage availability in this kind of world?

It is important to understand the availability requirements and characteristics of the service that is being provided or consumed, not just the system. For example, if you need to ensure 7x24 availability of a "Credit Authorization

Highlights

Service," then this includes not only the systems or the application but the entire service as defined by the business process. Since an SOA consists of services delivered by loosely coupled applications and systems that can potentially interact within and across organizational boundaries, it is critical to be able to deal with events when they occur in the various infrastructure components that affect the availability of those services.

This is where technologies come into play that monitor and sense faults, and process those events through an event management system. Here policies have been pre-defined to analyze these events, with a plan or course of action to address the failing component through some change to the environment. Once the appropriate plan is determined, it is automatically executed to repair the fault and maintain service availability. A repair example could be to bring another resource online, isolate and fence off the failing resource, and provide an alternate path to the new resource so that the service consumer can still get their service request processed. This notion of monitoring, analyzing, planning, and executing can occur down at the lowest levels of the infrastructure, but is part of a hierarchy that includes software, middleware, application, and business process layers. IBM calls this autonomic computing.

Using autonomic computing techniques to support SOA systems can help provide intelligent, adaptable infrastructure to increase availability.

Autonomic computing is the ability of systems to be more self-managing. The term "autonomic" comes from the autonomic nervous system, which controls many organs and muscles in the human body. Usually, we are unaware of its workings because it functions in an involuntary, reflexive manner -- for example, we don't notice when our heart beats faster or our blood vessels change size in response to temperature, posture, food intake, stressful experiences and other changes to which we're exposed. IBM's vision of autonomic computing embraces the development of intelligent, open systems capable of running with minimal human intervention, adapting to varying circumstances in accordance with business policies and objectives, and preparing these resources to help ensure sustained availability by automating recovery responses.

Consideration needs to be given to put the appropriate infrastructure in place (hardware and software and policies/plans/processes) to be able to sense/capture the failures, and provide the plans and resources to dynamically address those failures so that the SOA remains robust and continues to deliver service, and that unscheduled or scheduled (for repair actions) outages are kept to a minimum or eliminated entirely. This is particularly important in SOA since the notion of using distributed business partners to provide and consume services means that unavailability of those partner services may result in business impact across organizational boundaries.

Highlights

In this case, we need to ensure that appropriate monitoring, measuring and reporting capabilities are in place to know the availability of services in a SOA (be it your own or a business partners) and a mechanism to understand impacts to availability when changes to the architecture are proposed or injected by any party involved in the service delivery. Availability in a SOA is not just about dealing with service failures but also about configuring and managing the business process to deliver the service at the appropriate service level (which typically includes performance and availability metrics).

Lastly, availability in the SOA world is about making sure that the systems have the right levels of availability by taking advantage of hardware and software reliability technologies within the servers and storage using spare and fault tolerant components. This item is no different than availability solutions used today. What is relatively new is providing levels of availability through the use of orchestration, provisioning and virtualization techniques to provide virtual computing resources from among a set of physical machines and using those virtual servers or storage to provide the service when another has been removed for maintenance or because of a failure.

Virtualization

In a SOA, virtualization becomes a key enabler for delivering the necessary computing resources in a flexible dynamic manner when services are being requested and deployed. Virtualization provides the ability to quickly and efficiently utilize hardware and software resources in order to establish computing capability based on policy and workload characteristics. But what is it, in "real world" terms?

The use of virtualization of computing resources to support SOA applications can help provide flexibility, responsiveness and optimization of the IT system resources.

Virtualization is about providing a layer of abstraction between the computing resources and those that want to use them. It provides a logical rather than a physical view of data, computing, network and other resources available to applications and users. For your IT teams, virtualization lets you share resources - like servers, software and storage - to increase business flexibility, improve service levels and help reduce costs. IBM has been a leader in virtualization technologies for more than 35 years. We believe that our approach to virtualization enables IT to be more responsive and flexible to the changing needs of those adopting SOA.

For example, in a SOA application you may wish to execute a specific choreographed business process that makes use of capabilities in the existing infrastructure but requires additional resources due to new workloads in the choreography that were not present when a different path in the business

Highlights

process was taken. A scenario might involve executing a credit authorization when a specific rule is triggered due to a threshold being reached in a mortgage application. This may suddenly require more application serving capacity than was used during normal business processing, and so additional resources may need to be brought to bear.

These additional resources could be added using virtualization. Various techniques include:

- Dynamically adding Web application servers in a WebSphere/XD environment
- Addition of more capacity to a logical server running in a partition on the fly
- Addition of another logical server in its own partition on a physical computer

But it should be noted that virtualization is not only applicable to computing resources but to other IT resources as well, such as storage virtualization, network virtualization, and others.

Several key technologies can enable virtualization capabilities in the IT infrastructure.

Many people equate virtualization with partitioning. It's a natural response from a generation of business-computing users who've created and shared virtual workloads and devices with the IBM mainframe's PR/SM and virtual machine (VM) hypervisors and/or VMware workstations. What follows is an introduction to some of the key technologies that enable virtualization capabilities.

- **Dynamic logical partitioning (dynamic LPAR)** enables selected system resources like processors, memory and I/O components to be added and deleted from dedicated partitions while they're actively in use.
- **Micro-partitioning**, which was introduced with the POWER5 microprocessor-based systems, allows administrators to more efficiently allocate server resources by assigning resources to fractions of processors as opposed to whole processors. Resources can be shared easily, and changes in resource allocation are transparent to users.
- **Virtual I/O** is a special-purpose partition that provides virtual I/O and network resources to client partitions. The Virtual I/O Server owns the resources that are shared with clients. A physical adapter assigned to a partition can be shared by one or more other partitions, enabling administrators to minimize the number of physical adapters they require for individual clients. The Virtual I/O Server can thus reduce costs by eliminating the need for dedicated network adapters, disk adapters and disk drives.

Highlights

- **Virtual LANs**, which are configured through software rather than hardware, allow computers to behave as if they are located on the same LAN segment, even though they may be physically located on different segments. Extremely flexible virtual LANs help provide virtualization capabilities by allowing you to prioritize traffic on shared networks. A virtual LAN is organized by traffic patterns rather than physical location.
- **Virtual Ethernet** communication paths are provided between multiple operating systems (e.g., AIX 5L and Linux) without the need for any additional hardware. Virtual Ethernet segments can be dynamically created and access to a virtual LAN segment can be restricted for security or traffic segregation requirements.
- **HiperSockets** provide an integrated virtual TCP/IP network connection among multiple combinations of virtual servers and LPARs running Linux on zSeries, Linux on S/390, z/OS V1R2, z/VM V4R2 and VSE/ESA V2R7 or later on zSeries.
- **SAN Volume Controller** is designed to enable changes to physical storage with minimal or no disruption to applications. It combines the capacity from multiple disk storage systems into a single storage pool, which can be managed from a central point. SAN Volume Controller also allows you to apply advanced copy services across storage systems from many different vendors.
- **SAN File System** is designed to help reduce the associated costs and complexity of managing files within SANs by providing centralized and policy-based storage and data management for supported heterogeneous server, OS and storage platforms.
- **IBM Director** provides a single, Java-based user interface for monitoring the usage and performance of heterogeneous system resources (including Windows®, AIX 5L v5.3, Linux on Power, Linux on zSeries® and VMware configurations, as well as storage devices), as well as tracking inventory and events. Administrators can manage tasks, take core corrective actions and issue distributed commands.
- **Enterprise Workload Manager (EWLM)** provides a consolidated end-to-end view of transactions and their performance across a heterogeneous infrastructure as defined against business goals. Select systems resources, such as POWER5 partitions, can also be managed with EWLM.
- **Resource Dependency Services (RDS)** tool automatically discovers IBM and non-IBM resources, both physical and logical, and maps the virtualized topology, associations, and dependencies of any customer defined environment.

Summary

Highlights

Most IT professionals agree: buying computers is easy, building an infrastructure is hard. With the advent of SOA and Web services, the infrastructure becomes even more important in the overall solution. With a distributed application model such as SOA and Web services, infrastructure capabilities such as security, management, connectivity, performance, integration, high availability and architecture are a key cornerstone to a mature SOA enterprise.

Infrastructure must evolve to support Web services and SOA solutions. Of these capabilities, IT security and service management are two of the critical enablers for businesses to support SOA implementation. Without these capabilities, the solutions won't withstand production readiness.

The adoption of SOA is a watershed event for IT infrastructure transformation.

The IT transformation is being led by application simplification and componentization. This in turn is helping drive the transformation of infrastructure to support SOA and Web services. Application simplification towards Web services and SOA is a watershed event for IT transformation, including data centre infrastructure, operations, hardware and software.

Next generation IT architectures are focusing on technologies that help simplify the enterprise and support Web services and SOA. Techniques such as virtualization, resource provisioning and orchestration, and increased automation all assist in providing new ways to allocate and manage finite resources to support SOA. Businesses want to do more with less and simplify their existing environments. How do you begin to transform your IT infrastructure and operations to support SOA? This is where IBM is stepping up to help you address this need.

IBM Infrastructure Services for SOA

From strategy to implementation: IBM offers a comprehensive set of services to help you plan, design and manage an SOA environment

Integrated Technology Services (ITS) has announced that three of its on demand infrastructure services have been enhanced to globally support SOA and Web services. ITS has updated its services and knowledge base to include SOA and Web services in the full range of infrastructure services that ITS provides – from strategy and planning to design and implementation.

ITS is equipped to cover the core infrastructure services that an SOA deployment affects such as IBM WebSphere, Microsoft.net services, integration,

Highlights

testing, security, and SOA management. Typically SOA management is the most immediate infrastructure need. Combining its extensive experience with ITIL and enterprise systems management with its expertise with key software vendors in this space like Tivoli and AmberPoint, ITS is well prepared to help customers with their SOA management needs. ITS is also uniquely positioned to help customers more quickly and effectively deploy SOA applications and Web services by addressing infrastructure requirements with proven reference architectures and designs. Tight integration with both our WebSphere and Enterprise Services for Microsoft Technologies teams enables smooth and professional deployments, regardless of the infrastructure platform selected for SOA applications.

ITS has updated three on demand infrastructure service offerings to take advantage of the new architectural approaches and technologies that SOA and Web services bring to an infrastructure design. These Infrastructure Services for SOA include:

- **Infrastructure Services Readiness Engagements (ISRE)**: This offering provides an overall approach for determining the preparedness of your IT infrastructure to meet current business challenges and position for future on demand opportunities such as an infrastructure to support an SOA. These services produce actionable recommendations and solution initiatives in the form of IT transition roadmaps governing the adoption of infrastructures to support a Service Oriented Architecture. Infrastructure services readiness engagements evaluate IT processes, organization and technology utilizing best practices models for an on demand operating environment. We identify key areas for improvement and provide recommendations on how to transform your IT infrastructure and leverage on demand capabilities to support the implementation of SOA.
- **IT Service Management Design (ITSMD)**: The IT Service Management Design offering is an integrated set of planning and design engagements conducted within a rigorous and well-documented methodology. They feature a full life-cycle approach, best practices, industry process frameworks and open-standards such as ITIL (IT Infrastructure Library). Many clients seek an IT management model that is highly automated, flexible and able to quickly adapt to marketplace changes and business requirements. This service can help create a comprehensive framework and infrastructure management capabilities to support on demand requirements through a service-oriented, demand-driven IT management strategy and architecture, and accompanying delivery capabilities.

Highlights

- **Infrastructure Services Architecture Design-Enterprise (ISAD)**: This service assists you in developing an architectural framework and infrastructure designs to support a Service Oriented Architecture. It utilizes proven reference architectures and leading practices to help accelerate the SOA design process incorporating state-of-the-art technologies into your existing environment. This offering helps accelerate the design process integrating on demand technologies into existing IT environments and creates a roadmap that defines tactical projects to transition to an open, integrated, flexible and affordable infrastructure to support SOA. This service offering addresses specific design models for applications, infrastructure, data and security features that incorporate new technologies for integration, virtualization and automation.

In addition to the Infrastructure Services for SOA, IBM also has a wide range of service offerings available to assist with component business modelling and decomposition through to SOA application design and deployment. No matter what stage you are at in your adoption of SOA, IBM has service offerings to help accelerate your projects by leveraging our deep expertise.

Conclusion

A successful SOA implementation includes both a SOA application as well as an IT infrastructure ready to support SOA and Web services.

While the move to adopt SOA by many businesses is being spearheaded by their application development groups, the corresponding infrastructure and operations teams must also be involved to ensure that a production ready SOA infrastructure is in place to support their newly designed applications. Simply put, the formula for a successful SOA implementation equals SOA applications plus an SOA infrastructure. This will ensure that requirements and expectations are met in terms of overall service delivery by both components.

If you're a company that wants to grow, service orientation is one of the ways to do it, because it allows you to more flexibly link into your business partners, make changes, and respond to the market. Service Oriented Architecture is the DNA for all on demand businesses. So if your goal is to become one, then you really can't reach the ultimate potential of that goal without the flexibility that a Service Oriented Architecture brings to the table. By enhancing the infrastructure to support SOA, you can change how you typically provide IT services to your applications and to the business as a whole – by moving from a systems to a services focus.

Information Technology Services
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
04-06

X05-2010-00