HP OpenView Service Oriented Architecture Manager

Concepts Guide

Version: 2.11

Windows, HP-UX, Linux

Aug 2006

© Copyright 2004-2006 Hewlett-Packard Development Company, L.P.

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2004-2006 Hewlett-Packard Development Company, L.P.

Trademark Notices

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Linux is a U.S. registered trademark of Linus Torvalds

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation

UNIX® is a registered trademark of The Open Group

Support

You can visit the HP OpenView web site at:

http://www.hp.com/managementsoftware/support

This web site provides contact information and details about the products, services, and support that HP OpenView offers.

HP OpenView online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit enhancement requests online
- Download software patches
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and log in. Many also require a support contract.

To find more information about access levels, go to:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

http://www.managementsoftware.hp.com/passport-registration.html

Table of Contents

Glossary

1

Introduction

This chapter covers general information about this guide as well as overview information about the HP OpenView Service Oriented Architecture (SOA) Manager software. The information is presented at a high level and is not intended to address detailed aspects of the software. Such details are covered in subsequent chapters. The overview information includes:

- A technology primer
- Fundamental descriptions of the SOA Manager software
- A quick view of the problem space that the SOA Manager software addresses
- An overview of the SOA Manager solution

Document Overview

The SOA Manager Concept Guide provides a broad range of information that is used to understand the SOA Manager software and the context in which the software is presented. The content ranges from basic definitions for commonly used terms to advanced management concepts. The guide covers SOA and Web services concepts as required, but it is not meant to provide a detailed view of these technologies.

Read the guide sequentially starting with this chapter as each chapter builds on information presented in the previous chapters.

➤ If possible, read this guide prior to installing or setting up the software.

Audience

The Concept Guide is primarily intended for:

- Solution Architects that are positioning an SOA Manager solution
- Enterprise Architects that are evaluating an SOA Manager solution
- Operations Support Managers and administrators who have to support Web services based applications
- General technologists interested in the SOA Manager software

Prerequisites

The audience members for this guide should have a fundamental knowledge of Web services. Skill levels may range from basic Web service knowledge and implementations to experienced Web services implementers. General knowledge of .NET, Java, as well as software management principals is also helpful.

Technology Primer

This section provides an overview of several important technologies that are essential to the SOA Manager software. The technologies include: Distributed Management, Web services, and SOA.

The information provided in this section is meant for anyone who is not familiar with these technologies. The information provides a quick overview of the technology and is not meant to be exhaustive. If you would like to learn more, a great deal of information about these technologies is available on the Internet. You can skip this section if you are already familiar with these technologies.

Distributed Management

Distributed management is an approach to managing resources that are deployed and distributed across an enterprise network environment. The role of distributed management is to provide various aspects of management such as monitoring, configuration, provisioning, fault detection, and performance to ensure greater reliability, scalability, and efficiency of the resources. There are many distributed management solutions that are used to manage just about everything in the enterprise– from systems and network infrastructure to distributed applications and business processes.

The SOA Manager utilizes many distributed management principles. The management architecture and the management capabilities of the SOA Manager are discussed later in this guide. However, there are some general architectural components that are common to distributed management-based applications. These components include:

- **Management Agents**: Management agents are software components that get installed on a computer and are responsible for performing management tasks. The agents are specific to what technology is to be managed on the computer. For example, there are different agents for hardware platforms, operating systems, databases, and applications–just to name a few. The management agents perform management tasks on behalf of a central management server.
- **Management Server**: A Management server is a centralized software component that aggregates the data that is gathered by any number of management agents. As implied, the agents and management server are able to communicate with one another.

  The management sever typically contains a management console that is used to view the management data in some meaningful context. For example, an administrator may use a console to view a graph of how well an application is performing or maybe to view a network map that shows the status of all the servers in a network.

  Sophisticated management applications may allow administrators to change settings on a remote computer from the management server as well as set up policies that automatically initiate resolution procedures when certain problems arise.
- **Management Proxies**: Management proxies are software components that get installed on a computer and are responsible for gathering management data for computers that do not have a management agent available to them. Like management agents, proxies communicate their management data to the management server. Management Proxies are popular because of their ease of deployment and their separation from the computers that are to be managed.

Web Services

A Web service is the means by which software is described, discovered, and invoked over the Web. The software can perform a service as simple as converting time zones or as complex as an order processing system that interacts with any number of additional sub-systems in order to fulfill a request. The most widely recognized types of Web services are those that utilize the following set of open standards:

- ***XML*** (Extensible Markup Language): XML is a markup language used to describe data and does not include any presentation logic for the data. XML, in regards to Web services, allows software applications to exchange data in a simplified and standard manner.
- ***WSDL*** (Web services Description Language): WSDL is an XML-based language that is used to describe Web services. The description includes, among other things, the operations that are used to interact with the software. In practice, the information provided in a WSDL file is the only information that an application client needs to know in order use a particular Web service.
- ***SOAP*** (Simple Object Access Protocol): SOAP is an XML-based protocol that is typically used over HTTP to send messages (commonly referred to as SOAP messages) between application clients and servers. SOAP is platform-independent and therefore can be used to exchange messages between dissimilar systems that use different programming languages. For example a software component that is implemented in C++ could be accessed by a Java client if both platforms support SOAP.

- ***UDDI*** (Universal Description, Discovery, and Integration): UDDI is a standard for creating a registry of Web services. The registry can be public for all to use or private for a company's own internal use. A commonly used analogy for UDDI is that of the Yellows Pages of Web services. People or applications use UDDI to discover Web services. Of all the previously mentioned standards, UDDI is not required to create or use Web services.

Many middleware application server vendors support Web services based on the above standards. These vendors include, but are not limited to, Microsoft .NET, BEA WebLogic Server, IBM WebSphere, and Sun Java System Application Server. There are also several open source alternatives (e.g., Apache Axis) to these commercial offerings. Generally speaking, a majority of the Web services are implemented either using the .NET framework or as Java Web services running in a J2EE compliant server. The SOA Manager software can manage both types of Web services.

Service Oriented Architectures

Service Oriented Architecture (SOA) is a set of principles that define an architecture that is loosely coupled and comprised of service providers and service consumers that interact according to a negotiated contract or interface. In every definition of SOA, there are some key notions of loose coupling between a service producer and consumer.

The ***Service Consumer*** is only concerned about what value the consumed service provides. Every service consumer has some need and can search for a service to fulfill that need. The service consumer determines in what context the service is used, and is not concerned about how the service is fulfilled.

The ***Service Producer*** focuses on how the service provides value and which resources provide the service. A service producer makes itself available for discovery and fulfills the consumers need, but the details of service implementation are abstracted or hidden (e.g., it may have few or many steps, can delegate to multiple resources, etc…). The underlying service implementation can change transparent to the consumer. The "dividing line" is an agreed upon ***Contract*** (or interface) that defines all the interaction between the consumer and the provider.

These concepts are applied to a business problem or IT environment to increase the potential for simplification, standardization, modularization, and integration. SOA is not a new concept and has been attempted with other technologies; however, Web services are currently the popular choice for enabling SOA implementations because of their inherent alignment (XML, WSDL, SOAP, UDDI) with SOA principals. The SOA Manager is focused on managing Web services-based SOA implementations.

What is SOA Manager

There are a number of ways to describe the SOA manager software. Each description is equally important and helps define what the SOA Manager software brings to an IT environment in terms of functionality and benefits. This section focuses on the most relevant descriptions, which are discussed in much more detail throughout this guide. The descriptions include:

- Management for the Adaptive Enterprise
- Business Service Management
- SOA Resource Management
- SOA Management Integration

Management for the Adaptive Enterprise

Management for the adaptive enterprise is a key HP strategy that is aimed at transforming IT into a business-focused, service-oriented organization. The SOA Manager software is part of this strategic initiative and focuses primarily on delivering the strategy in the area of SOA.

The strategy can be summarized as follows:

- Integrate people, process, and technology to run IT as a business.
- Automate the dynamic link between business and IT.
- Shift IT investment from maintenance to innovation.

This is the highest-level description of the software and describes one of the overall design principals. The strategy is evident in the software's functionality and unique management perspective.

Business Service Management

The SOA Manager software is used to define, manage, and integrate the lifecycle of business services. This is a practical description that is immediately evident when using the SOA Manager software.

Business services are used to better link IT organizations with Business organizations in order to increase business agility. In the context of the SOA Manager software, a business service is the virtualization of any business application that is offered by a business manager to either internal or external customers.

Business services are the main context of the SOA Manager's service model and a distinguishing characteristic from other management solutions. Business services and the service model are described in much more detail in Chapter 3.

SOA Resource Management

The SOA Manager software is used to manage SOA resources to ensure their reliability and optimal performance. This is the most concrete and practical description of the software.

Currently, the managed SOA resources that are supported include: Web services, Web services containers, Web services intermediaries, Databases, Message Oriented Middleware (MOM), and Globus-based Grids. These resources are defined within the context of an IT service and bound to a business service. These resources are the technical components that are used to deliver a business service. SOA resource management is described in much more detail in Chapter 2.

SOA Management Integration

The SOA Management software is an integration platform used to create unique management solutions that allow reuse of current enterprise management investments. This is a very low-level description of the software and is only relevant when customizing the software or performing solution integrations. Much of the integration is achieved using the SOA Manager's management interfaces that are exposed as Web services. Management integration is described in much more detail in Chapter 4.

Defining the Problem Space

The adoption of Web services and SOA in the enterprise application space is rising at a rapid rate. The acceleration is based on two benefits:

- Improved cost efficiencies through significant software reuse.
- Increased agility and adaptability provided by the ability to rapidly compose new business processes from existing services or to modify existing business processes.

However, there are significant challenges that must be addressed in SOA implementations for these benefits to be truly achieved:

- Web services and SOA exist in a dynamically bound and loosely coupled environment. In such an environment, problems occurring in a service implementation can impact multiple layers of consumers. Problem isolation and impact analysis become extremely difficult, thereby making it difficult to implement and abide by Service Level Agreements (SLA).
- Service implementation must address security, identity, access, and trust management issues.
- SOA policy definition and enforcement (better known as SOA governance) must be achieved at the enterprise level.
- Service deployments can quickly grow and evolve in order to achieve true agility. Service lifecycle issues require far better coordination and communication between business sponsors, stakeholders, and various groups in IT such as IT development, support, and operations.
- Many enterprises already have heavy investments in existing enterprise management solutions and processes. Emerging infrastructure and management implementations in the SOA space (e.g., Web services Management (WSM), XML firewalls, UDDI Registries, etc…) must be integrated with these existing solutions and processes.

These obstacles are categorized into a problem space called ***SOA Management***. There is no single product that can be purchased and implemented that can address these various challenges. HP SOA Manager addresses these obstacles by leveraging SOA as the foundation for the enablement of integrated management solutions. It is clear that what is required in this solution space is ***Service-Oriented Architecture Management***.

Within the enterprise, SOA Management affects three distinct areas: Service Development, Service Architecture, and Enterprise Management. Figure 1-1 shows the relationship of these areas.

The ***Service Development*** area is where Web services are created and packaged as applications. Various tools and IDEs are used to create the services. Like all applications, the services go through development and testing cycles.

The ***Service Architecture*** area is where Web services are hosted and discovered. This includes Web service containers (e.g., .NET or J2EE-based containers), Web service Brokers, and UDDI registries.

The ***Enterprise Management*** area is where services, together with other enterprise resources, are managed. Many different products and tools may be deployed and various processes may be used to monitor the enterprise.

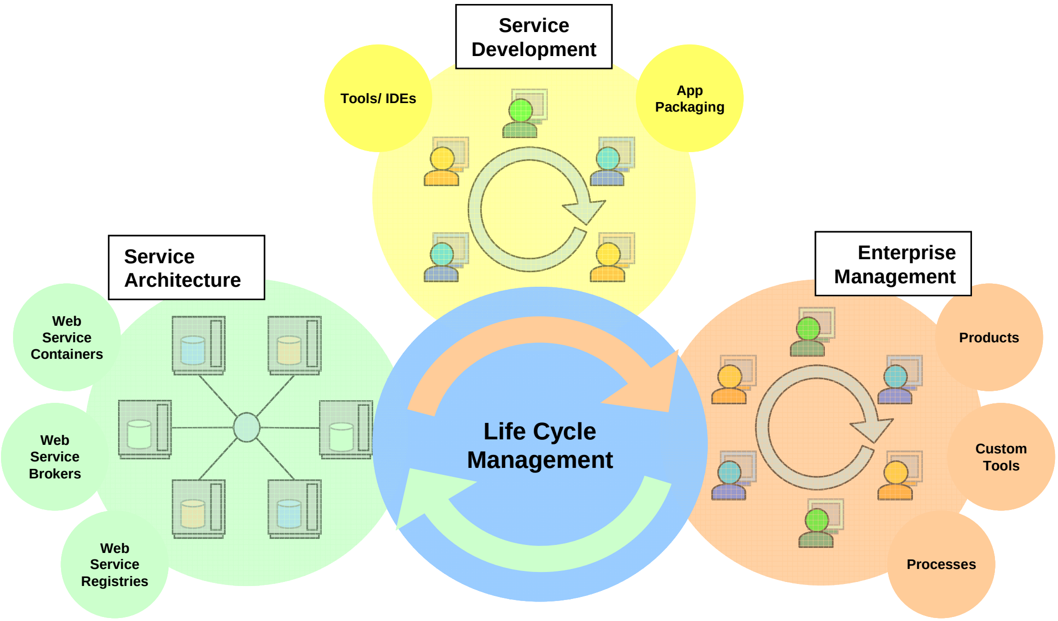

Figure 1-1: SOA Management Across the Enterprise

Solution Overview

The SOA Manager solution encompasses three broad areas of functionality to enable end customer solutions:

- Resources Management
- Services Model
- Enterprise Management Integration

Figure 1-2 shows these areas in relation to the SOA Management space. A brief description of these areas is provided in this section. Detailed documentation on these areas is provided in subsequent chapters.

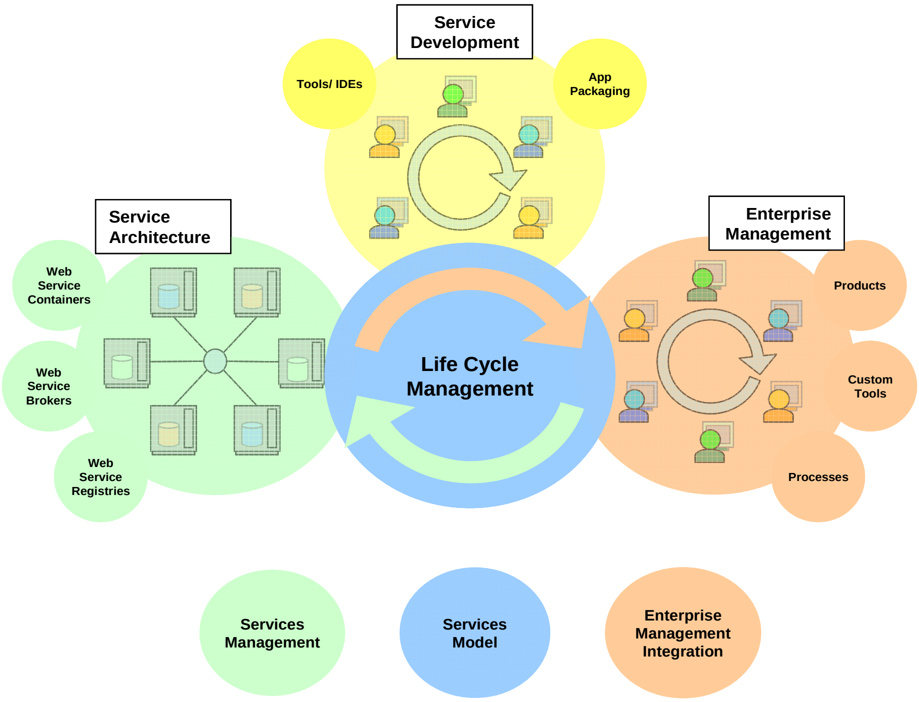

Figure 1-2: SOA Manager Solution Overview

Resources Management

The primary focus of resource management is on managing the resources that are part of an SOA environment. These resources expose their management capabilities through Web services interfaces using WS-based management protocols.

> The SOA Manager currently implements an HP-authored precursor of standard WS-based management protocols.

Figure 1-3 below shows Web services resources that are managed as part of resource management. See Chapter 3 for a complete list of resources that are managed by the SOA Manager. Since Web services currently make up the majority of resources in an SOA environment they are the main focus of resource management in this guide and are discussed in detail in Chapter 2.

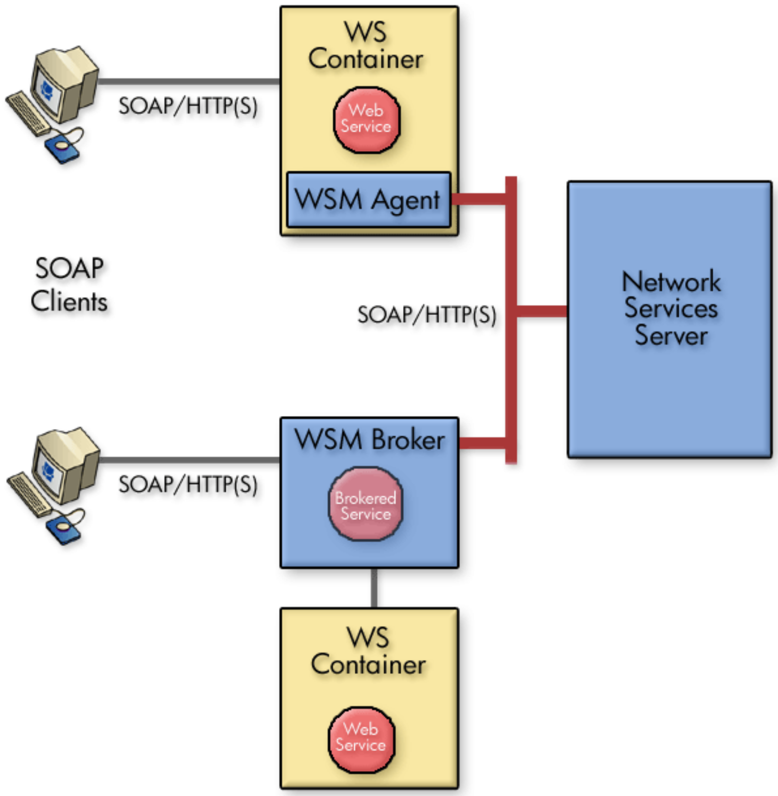

Figure 1-3: Resource Management for Web Services

Services Model

The SOA Manager is used to define and maintain a dynamic model of relationships between business services, their supporting software configurations, and the virtualized infrastructure. Figure 1-4 shows a view of the Service model. The Service model is discussed in greater detail in Chapter 3.

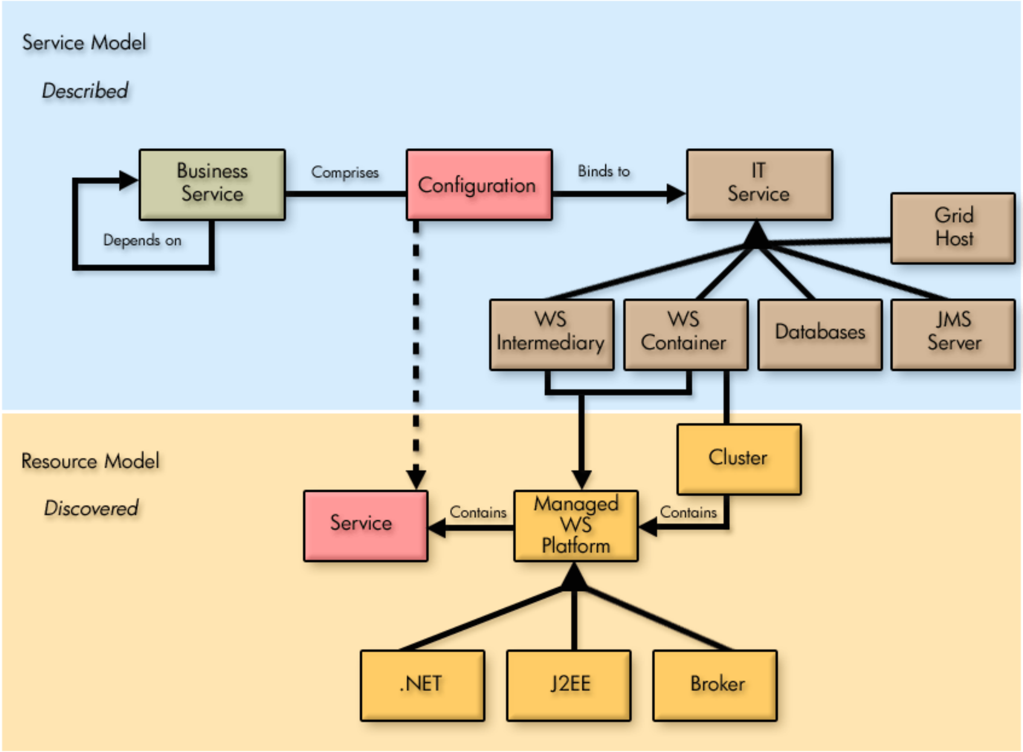

Figure 1-4: SOA Service Model

The goal of Management of the Adaptive Enterprise is to create a closely knit value chain of Business and IT resources so that business products are provided in an efficient and timely manner to either internal or external customers. This requires tight alignment between three functional groups within an enterprise: Business Managers, Application Development teams, and Application and Operations Support. This alignment is enabled by modeling the concept of a business service and providing the three groups with interfaces to interact with the service model.

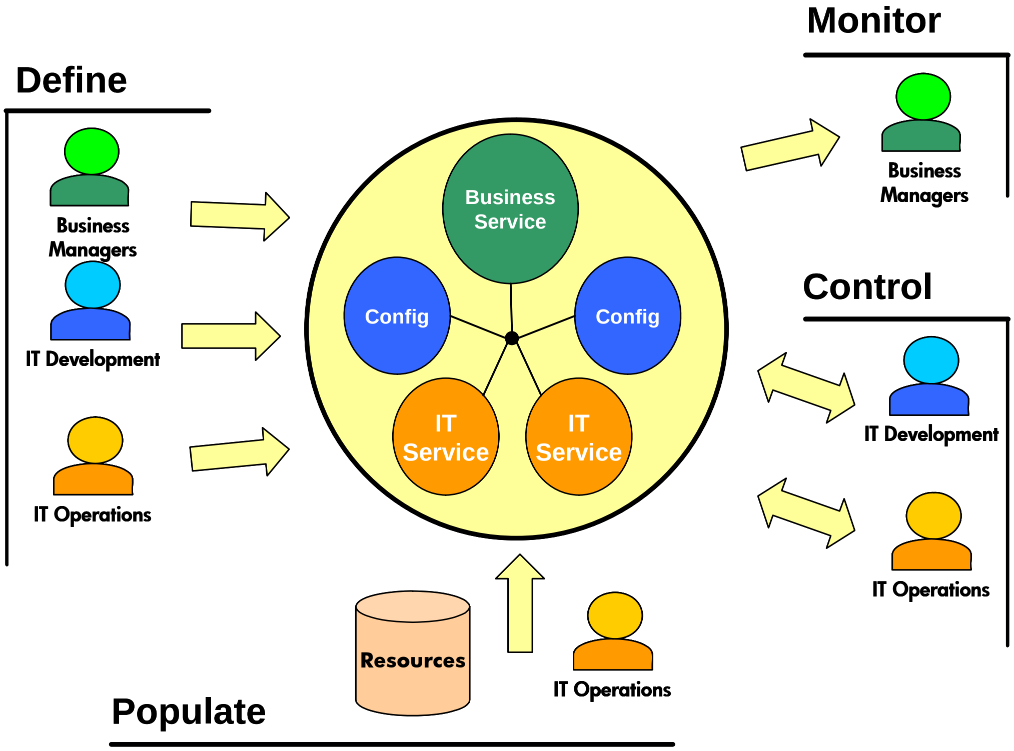

Figure 1-5: Service Model Roles

Enterprise Management Integration

SOA Manager provides a plug-and-play architecture that allows existing tools to be leveraged while augmenting HP SOA Manager's capabilities. Integration is typically completed in the following areas:

- Custom management functionality
- Integration with enterprise management products
- Reuse of SOA Management Web services and the SOA service model

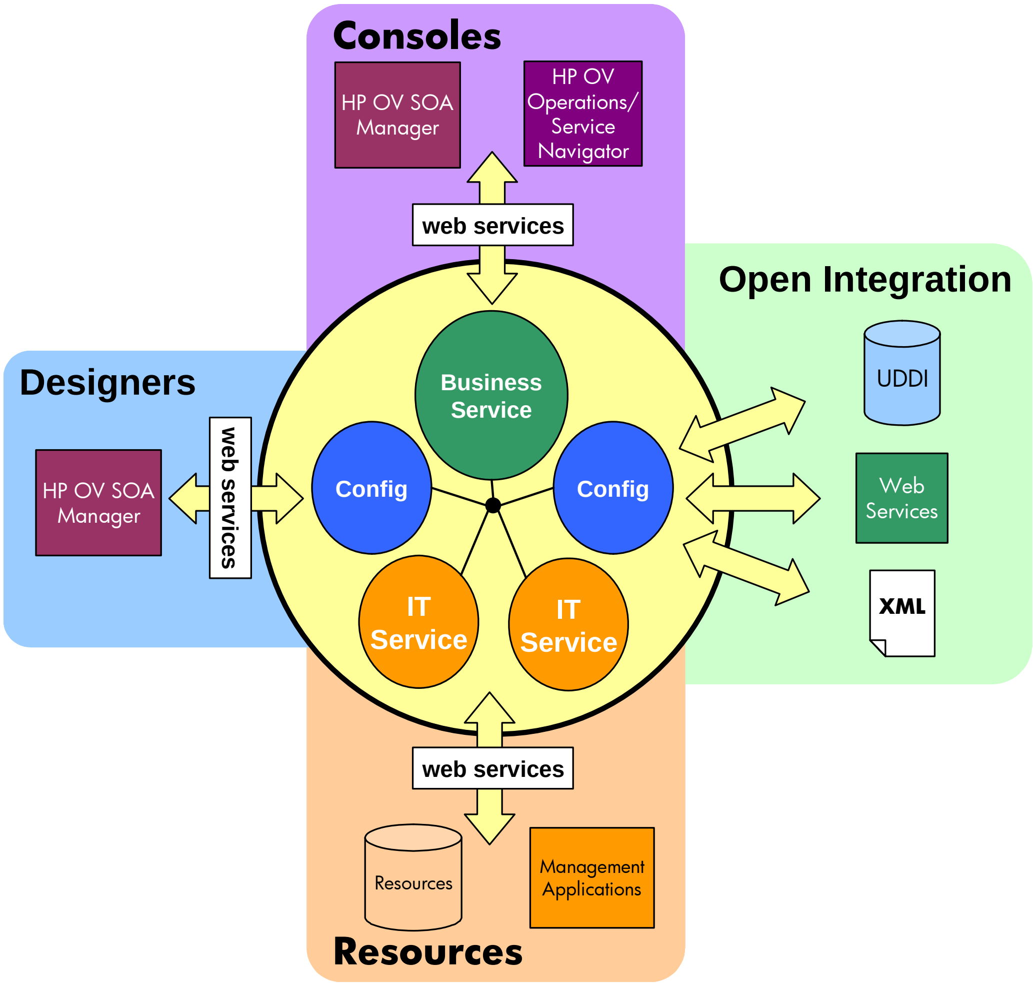

Figure 1-6: Enterprise Management Integration

This guide describes the integration architecture for management which is itself built on the following SOA design principles:

- Management infrastructure is modular. Management is provided by best-of-breed vendor products.
- Integration is standards-based. Integration is standardized by the adoption of the WS stack of specifications (such as WS-I, WS-Addressing, WS-Security, etc…).
- The Service's model represents the deployed infrastructure. This model captures various participants of the SOA and their relationships.

2

Resource Management

This chapter provides conceptual information about the SOA Manager's Resource management capabilities. In particular, the content focuses on Web services management. The chapter includes:

- **Overview**: This section provides a basic definition of resource management and why it is important.
- **Service Management Features**: This section provides a description of each service management feature and the benefits that each provides.
- **Enabling Manageability**: This section provides basic concepts that are needed to understand how the SOA Manager implements manageability for Web services and SOA resources. A generic architectural view is provided.
- **Deploying Manageability**: This section provides a description of the SOA Manager's Web Services Management (WSM) agents. These agents are installed in SOA environments in order to manage Web services and SOA resources.
- **Management Server**: This section provides a description of the default management server that is included as part of the SOA Manager's service management capabilities.

Overview

Resource Management is the act of managing the resources in an SOA that are being used by business applications. The SOA Manager includes a range of SOA resources that are vital to the success of a business application. These resources include: Web services, brokered Web services, Web service containers, WSM Brokers (a Web service proxy also referred to as a Web service intermediary), Databases, Grid Hosts, and Java Messaging Servers (JMS) including JMS Queues and Topics. This chapter only focuses on Web services management.

Operators and administrators utilize resource management to ensure SOA resources are always available and functioning within acceptable operating limits. When problems occur, resource management allows them to quickly identify possible causes and initiate appropriate resolution procedures.

Web services are prevalent within current SOA environment implementations and their management is essential to the overall success of an SOA. Web service management is important because the health and well-being of a single Web service may impact the overall health and well-being of multiple business applications.

WSM Features

There are some basic management objectives that the SOA Manager software provides in the service management space. These objectives include:

- Ensuring the high-availability of service-based applications
- Ensuring the optimum performance of service-based applications
- Decrease response and resolution times
- Maintaining a record of Web service usage
- Troubleshooting and diagnosing problems
- Ensuring the secure usage of Web services

The topics in this section describe the features that allow administrators and operators to achieve these objectives and more.

Availability Monitoring

Availability Monitoring determines when Web services and SOA resources are no longer operational and then generates an alert notification. This feature allows administrators to quickly react to errors and mitigate application downtime. Availability alerts are typically the first indication that a problem has occurred with a Web service or an SOA resource.

Performance Monitoring

Performance Monitoring captures a set of real-time performance metrics that clearly indicate the health, availability, and performance of Web services. The metrics include:

- Availability %
- Average Response Time
- Failure Count
- Maximum Response Time
- Minimum Response Time
- Security Violations
- Success Count
- Total Requests
- Uptime %
- Uptime

The SOA Manager software captures the performance and availability experience of real consumers, computed by monitoring real transactions, without doing externally probed synthetic transactions.

An important part of this feature is that the metrics are calculated over time. This allows operators and administrators to analyze changes in Web service performance. This is commonly referred to as ***Trend analysis***.

Impact Analysis

Impact Analysis is the ability to discover how the performance of a service affects other related services. When performing impact analysis in the SOA Manager, a relationship between services must be explicitly defined. For example, *Service A* is related to *Service B*. In this example, if *Service B* depends on *Service A*, then any performance problems for *Service A* can also affect *Service B*. Dependencies are defined in the service model.

Administrators and operators use this feature to quickly visualize the impact a poorly performing service may have on other services and business applications.

Root Cause Analysis

Root Cause Analysis is the ability to discover which services are causing a group of related services to degrade. Like impact analysis, a relationship between services must be explicitly defined. For example, *Service A* is related to *Service B*. In this example, if *Service A* depends on *Service B*, then any performance problems for *Service A* may be the result of a problem with *Service B*. Dependencies are defined in the service model.

Administrators and operators use this feature to quickly troubleshoot which service is causing an overall group of related services to degrade. A considerable amount of time can be saved by pinpointing a problem without having to manually complete a process of elimination.

SLO Monitoring

SLO Monitoring evaluates a Web service's performance metric values (described above) against pre-defined service level objectives (SLO). ***SLOs*** are the preferred operating limits for a Web service.

For example, an SLO may stipulate that a Web service's Availability % be greater than `90%` and have an Average Response Time of less than `200` Milliseconds. At runtime, if these SLO values are violated, an alert notification is generated that indicates the breach and also provides the actual values of the performance metrics (in this case, the actual Availability % and Average Response Time). An additional alert is generated when the service levels return to normal.

Administrators and operators use this feature to stay informed about changes in a Web service's performance before the changes can affect a user's experience or break a service level agreement (SLA). ***SLA*** is an agreement between a service consumer and a service provider about an expected level of availability and performance of a service.

Auditing

Auditing captures trace information for all Web service requests and responses. Trace information provides a historical record of a Web service's performance, access history, security, size, source and destination endpoints, successes, failures, and can also include the SOAP request-response payloads and profile data. Trace information can be persisted to a database at regular intervals. The information is used to generate audit reports or can be used by other auditing applications.

Administrators and operators use auditing for a number of reasons. The reasons may include repudiation (i.e., evidence or proof that an SLA has either been maintained or broken), billing or metering, or to validate unauthorized access to a Web service.

Content Monitoring

Content Monitoring searches Web service request and/or response messages for specific content. An alert notification is generated when the content is found.

Administrators and operators use this feature to react to events that can potentially have an impact on business operations. For example, when managing an order process service, an alert notification could be generated when:

- An important client is using the service
- An order total is greater than $25,000.00
- A specific product is ordered
- A specific product is shipped

Logging

Logging captures the local standard output for SOA components so that the output can be analyzed from a remote central location. Administrators and operators use this feature to view the current log messages as well as change log output levels in order to view more detailed log messages. Logging is particularly useful when troubleshooting problems.

Deployment

Deploying installs Web services or Web service proxies from a remote central location. Administrators and operators use this feature to install new or updated versions of a Web service. The feature is also used to install additional instances of a Web service or Web service proxies to compensate for increased demand. This feature can save administrators and operators a considerable amount of time and allow them to quickly adapt to changes in business applications.

Security

Security ensures that access to Web services is secure. The security features are implemented using several industry standard security technologies and Select Access. These include:

- Transport Level Security: HTTP/S, X.509 Client Certificates
- Message Level Security: XML Digital Signature, XML Encryption - WS-Security
- Authorization/Authentication: AAA Security Integration with HP OpenView Select Access

Administrators and operators typically use these features to secure communication to and from the SOA Manager's WSM Broker. The WSM Broker is discussed later in this chapter.

Enabling WSM Manageability

Most enterprises have standardized on a Web services container to host their Web services. Web services containers are included with standard J2EE platforms (such as BEA WebLogic, IBM WebSphere, Apache Axis, JBoss, etc…). Microsoft uses the .NET platform and the Internet Information Server (IIS) to host Web services. Lastly, there may be proprietary server environments that provide a means of hosting Web services.

None of these containers expose adequate instrumentation and manageability information about their hosted Web services. Moreover, the specifications used to implement Web services (such as WSDL and SOAP) do not provide any built-in manageability. Web services management vendors are left to their own devices to implement appropriate manageability instrumentation hooks into Web services.

The SOA Manager software uses ***Interposed Manageability*** to manage Web services. Interposed manageability means inserting management policies in the request/response path of Web services. This allows Web services to be managed in a standard and consistent manner and also allows the management of Web services that were not designed with manageability in mind.

Enablement Architecture

Management Policies contain the management logic that is used to interpose visibility and controls on Web services. The actual implementation of the management policies is done using ***Policy Handlers*** (also referred to as simply ***Handlers***). Handlers are inserted in the HTTP or SOAP pipeline that is responsible for processing request and response messages. Multiple handlers can be linked together in a ***Handler Chain***.

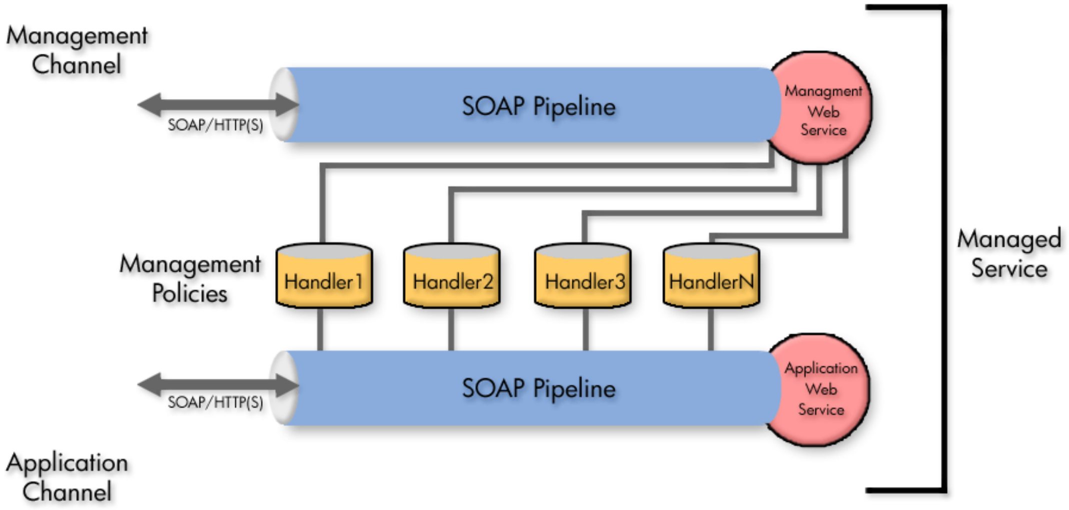

Figure 2-1: Generic WSM Enablement Architecture

The management information that is obtained from the handlers is then exposed through a ***Management Web Service*** that utilizes standard Web services management protocols. Management clients (such as the Network Service server) use the management Web services to construct views of management data. Management Web services provide both operations and event exchange patterns.

> Management Web services are also used to provide manageability for WS containers such as discovery and deployment of Web services to the WS container.

Management Web services can be published to a separate ***Management Channel*** instead of the ***Application Channel***, which typically only contains application related traffic. For example, an application could invoke Web services that are deployed in a Web services container which is accessed using port 8080. The Network Service server invokes the Web service's management Web service using port 8090.

Independent channels provide a good separation of management and application traffic to ensure that management traffic does not adversely affect the performance for users of the application.

Web Services Management Standards

There are several management specifications that are driving some standardization and adoption of Web services-based manageability protocols into the management of SOA deployments. Two such specifications are: Web Services Distributed Management (WSDM) and Web Services for Management (WS-Management). At present, the SOA Manager software utilizes the Web Services Management Framework (WSMF), which is an HP authored pre-cursor to the standard WS management protocols. The SOA Manager software will support these specifications as they mature and stabilize.

Deploying WSM Manageability

There are two ways to manage Web services: directly in a Web services container (using a WSM Agent), and through the use of a Web services intermediary process (using the WSM Broker). Both of these deployments are typical of distributed management solutions. In both cases, the underlying enablement architecture described in the previous section remains relatively the same. This section describes each of these deployment options. Service Management setup typically begins by setting up one of these options into a service environment.

WSM Agents

A ***WSM Agent*** is an enablement component that is installed in a WS Container in order to manage the Web services in the container as well as the container itself. WSM Agents are implemented using specific technologies and programming languages that are native to the WS Container's platform. There are currently two agents that can be deployed:

- The ***WSM J2EE Agent*** – an agent for the WebLogic Server Web services Container. The J2EE agent is written leveraging Java J2EE specifications and technologies such as JMX, JAXRPC, RMI, and Servlets.
- The ***WSM .Net Agent*** – an agent for the .NET Web services Container. The .Net agent is written as a .Net C# application that leverages Windows/.Net technologies such as WMI instrumentation, global HTTP pipeline and the WSE pipeline offered by the .Net programming model.

Agents are often referred to as ***Platform Native WSM Agents*** because they are specific to a certain platform. Figure 2-2 shows a high-level architecture of a WSM Agent.

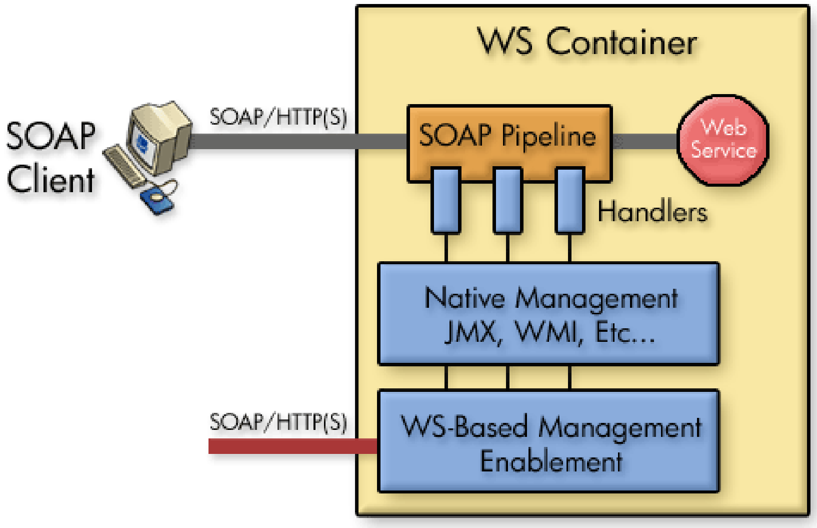

Figure 2-2: WSM Agents

The agents have two core components:

- A set of policy handlers that must be incorporated into the Application Web service path. The handlers update JMX or WMI instrumentation internally. SOA Manager ships with the implementation of these handlers. However, the handlers must be manually enabled by modifying XML files.
- A management-related Web services application that converts native JMX or WMI information into standard management Web services. This application is provided by SOA Manager but must be installed into a WS Container. This is an initial one time installation and configuration process.

Installation and configuration instructions for the WSM J2EE Agent and the WSM .NET Agent are located in the *WSM J2EE Agent Administrator Guide* and the *WSM .NET Agent Administrator Guide* respectively. These guide are located in the `/documentation` directory of the distribution.

WSM Broker

The ***WSM Broker*** is a flexible, configurable, high performance Java-based Web services Intermediary process. It is a self-contained application that runs in a single JVM process and does not require an existing Application Server. The WSM Broker is not specific to any particular Web Services container; therefore, it can be used to add manageability to any container.

The WSM Broker hosts ***Brokered Services***, which are Web service proxies to the Web service being managed.

A brokered service must be created for each Web service that you want to manage. The WSM Broker contains a tool called the Broker Configurator. The Broker Configurator includes wizards for creating brokered Web services and enabling/configuring the pre-defined handlers that are pre-packaged with the Broker.

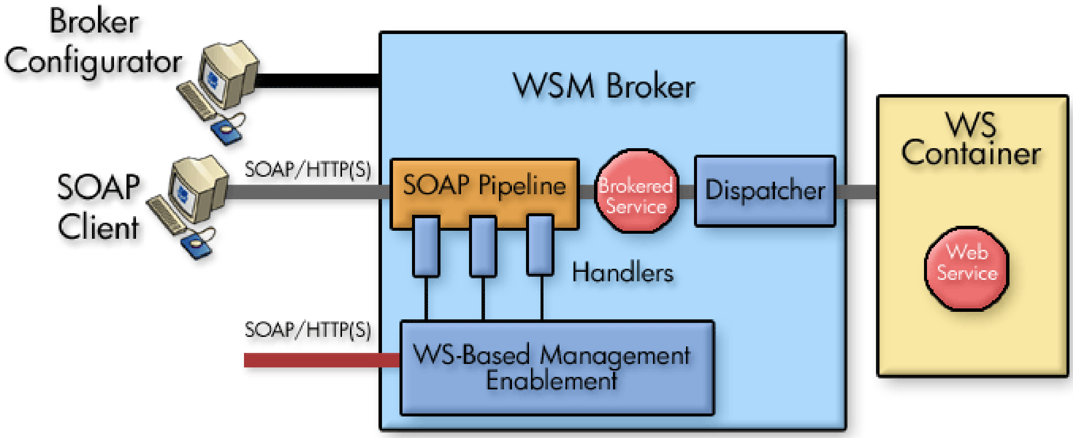

Figure 2-3: WSM Broker

Advanced solutions can create custom handlers and include them for execution in the Broker. Advanced solutions may also bypass the Broker Configurator tool and directly create configurations using the documented XML configuration format. The *SOA Manager Integrator Guide* contains detailed information about the Java-based APIs for creating custom handlers. The custom handlers cannot build additional information such as new metrics or any behavior that reflects in the SOA Manager Business Service Explorer console. However, custom handlers can do some local processing of messages.

The WSM Broker generates a WSDL for each brokered service, replacing the application Web service endpoint with the brokered Web service endpoint. Web service consumers send requests to this brokered Web service. Steps must be taken to ensure that consumers do not directly go to the Web service implementations and are always routed through the Broker. This can be done using manual processes and best practices or enforced using IP firewall policies.

Runtime requests that are sent to the brokered service are processed for management and security purposes, and then forwarded (dispatched) to the actual service's endpoint(s). The extra network hop introduces a small latency that is less than 20 ms for each transaction for very basic monitoring. The latency increases proportionally as you add other features such as security authentication and authorization. Due to the considerable flexibility and separation of concerns they provide, Web service intermediaries have gained considerable mainstream acceptance as a preferred means for managing Web services.

Supported Handlers for Agents and Brokers

As previously mentioned, several pre-defined Handlers are provided for both Agents and Brokers, including:

- **Performance and Fault Monitoring**: Capture response times and faults for messages to generate metrics for SLO monitoring.
- **Logging**: Log detailed diagnostic messages to local log files. Some other vendors describe Logging as the ability to log information to a central database. We describe this feature of capturing messages to a database as Auditing and refer to Logging as the capability used for SOA component troubleshooting rather than Web services application troubleshooting.
- **Auditing**: Collects context and payload of SOAP messages sent to Web services. The information collected is sent to the Network Services server and is stored in a database.
- **Business Content Alerting**: Lets you "watch" SOAP message payload (content) and raise alerts when certain conditions are met. For example, a business manager might want to be informed if someone with a credit rating of less than 5 applies for a loan. This is information that could be detected using Business Content Alerting.

Additional pre-defined security handlers are provided for the Broker, including:

- **Inbound Message Security**: Provides authorization using the principal and credentials associated with an operation. The authorization is done using a configured security provider such as Select Access.
- **Outbound Message Security**: Provides support for WS-Security on outbound messages (i.e., from the Broker to a Web Services container). This includes user name/password, signing, and encryption.
- **Schema Validation**: Validate that SOAP requests conform to a Web service's WSDL or reject the message and return a SOAP fault. This stops malformed messages from reaching the Web service implementation. This handler is not provided for Agents.
- **Security Auditing**: Used to collect security trace information (used for non-repudiation) and to send the payload to a security provider. For example, when using Select Access to control authorization, the traces can be viewed using the Select Access Audit Report Viewer.

No security Handlers are provided for the Agents because security is handled by modern Web Service containers for J2EE and .Net. Additionally most Identity and Access Management solutions offer native integrations into these platforms.

In addition to the pre-defined Handlers described above, you can add your own custom handlers to a handler chain. Custom Handlers are created using the SOA Manager APIs. For more details, refer to the *SOA Manager Integrator Guide*.

Management Server

The previous sections described how the WSM Components interposed manageability in order to gather management data about Web services and SOA resources. The final step of service management is to use a management server to collect the management data and make that data available in a meaningful context. There are three options that can be used to accomplish this:

- Using the Network Services Server and the Business Service Explorer (default implementation that is provided out-of-box)
- Using other enterprise management clients (requires integration)
- Creating your own management clients (requires integration)

This section discusses the first option. The last two options are discussed in Chapter 4, “Enterprise Management Integration.”

Network Services Server

The Network Services Server is a central management server that runs in a single Java Virtual Machine (JVM) process. A typical SOA Manager installation includes a single Network Services server that interacts with any number of WSM Agents and/or WSM Brokers that are deployed in an SOA environment. As previously discussed, the WSM components expose Web services and SOA resource management data as management Web services. This means the Network Services server itself is a SOAP client that utilizes SOAP/HTTP(S) to invoke the management Web services.

Figure 2-4 below shows the interaction and relationship between a WSM Agent, WSM Broker, and the Network Services Server.

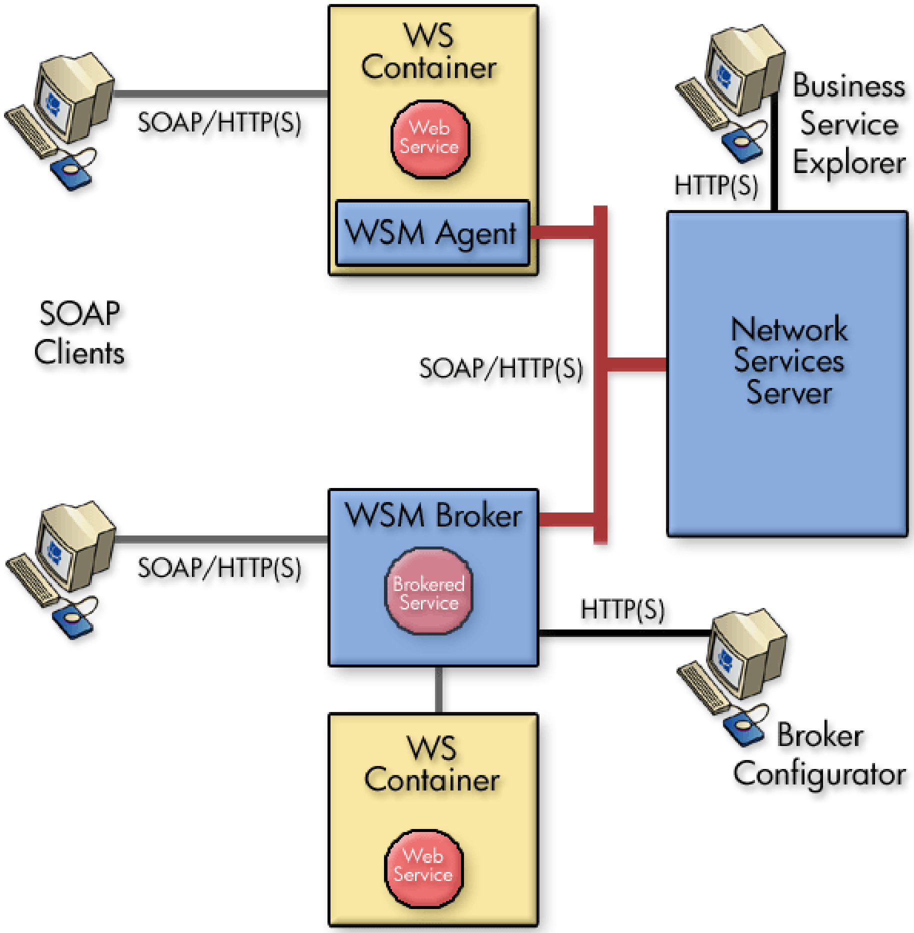

Figure 2-4: Deployment Architecture

Business Service Explorer

The SOA Manager's Network Service server includes a management/administrative console called the ***Business Service Explorer*** (BSE). The BSE, among other things, is used to construct a view of the management data that is being collected by the Network Services server. The BSE is also used to interact with and configure many of the service management features that are discussed in the "Service Management Features" section above.

➤ The Network Service server and the BSE are also used to model Web services and SOA Resources into the service models. The following chapter discusses the SOA Manager's Service Model capabilities.

The BSE is a J2EE-based Web application that is hosted within a light-weight Servlet container that is included with the Network Services server. The BSE is automatically installed as part of the SOA Manager installation and does not need to be manually deployed to the Network Service server. Once the Network Services server is started, the BSE can be accessed from any Web Browser. As the default, the BSE can be accessed on port 5002 of the Network Services server's host computer.

3

Service Modeling

This chapter provides conceptual information about the SOA Manager's service modeling capabilities. The information includes:

- **Overview**: This section provides a basic overview of a service model and the importance of using service models.
- **Conceptual View**: This section provides a description of each element in the service model and how the service model relates to different individuals in an organization.
- **Defining Service Models**: This section provides a summary of the steps that are used to define service models when using the BSE.

Overview

A ***Service Model*** is the virtual representation of managed SOA resources. Currently, these resources include: Web services, brokered services, Web service containers, Web service proxies (also referred to as Web service intermediaries), Databases, Java Messaging Servers (JMS), and Host services that are built using the Globus Toolkit-based grid.

The service model's structure provides an organized view of the managed SOA resources and their relationships to each other. The structural elements that make up the service model are:

- Business services
- Configurations
- IT services
- Application resources

These structural elements are detailed in the "Conceptual View" section below. The section primarily focuses on an end user's view of the service model. However, at the code level, the service model is represented as a management information model, which is exposed externally in order to create integrated management solutions. The model can also be published to UDDI and reused by other management applications. Integration is discussed in the following chapter.

The use of service models offers several advantages over more traditional management conventions. The advantages are listed below:

- A service model provides a more relevant management context for managing SOA resources. The context goes beyond simply managing Web services to include all resources in an SOA environment. These resources are viewed from a vantage point relevant to both business personnel as well as IT personnel. The service model inherently links both these groups.
- A service model allows Business Managers, Developers, IT Operations, and Support personnel to collaborate when defining, creating, deploying, and managing SOA resources.
- A service model allows for the automation of common tasks that are typical of SOA implementations such as the discovery of managed resources, the deployment of managed resources, and endpoint routing.
- A service model can remain valid even as the managed SOA resources are updated and changed.
- A service model allows the management information of SOA resources to be reused by other applications and facilitates enterprise management integrations.

Conceptual View

The service model is comprised of two main structural elements: IT services and business services. This section describes these structural elements in both abstract terms as well as their specific application in the area of SOA. The description also includes the roles various people in an organization play in relation to these service model elements.

Figure 3-1 below shows the service model elements starting with a business service. Business services are the entry point of the service model.

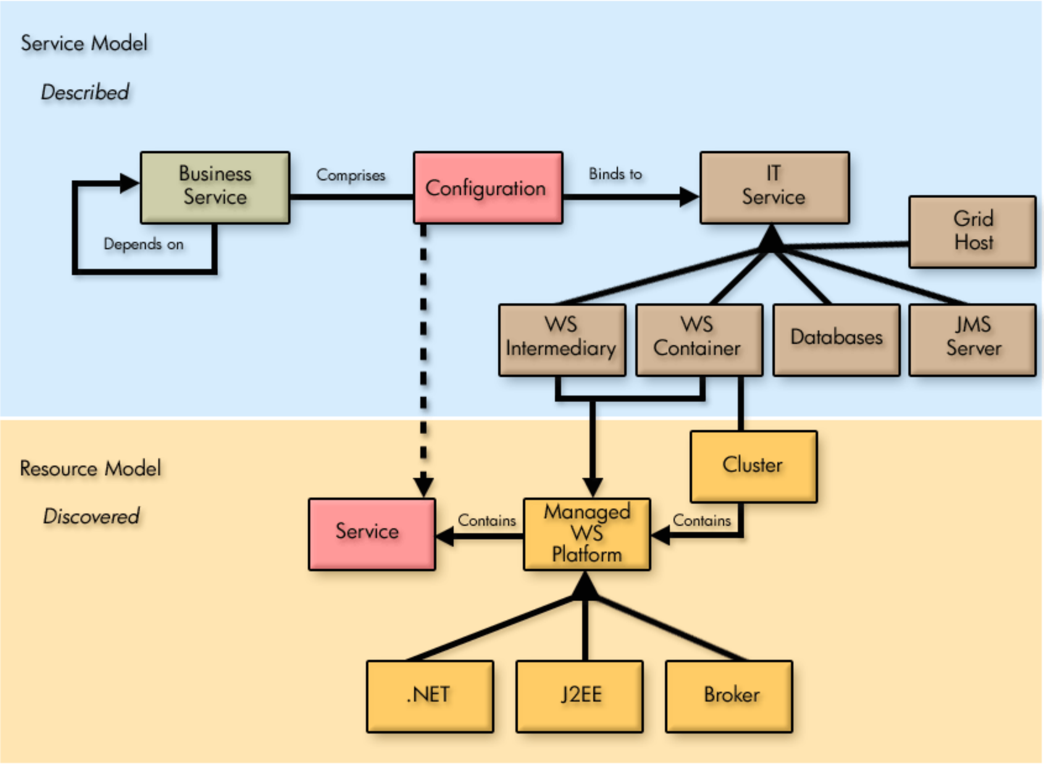

Figure 3-1: Service Model Elements

IT Service

An ***IT Service*** as captured in the SOA Manager represents the virtualization of management information or capabilities of a group of resources of a certain type that are associated with a set of stakeholders. The concept of virtualization of IT resources for the purpose of consumption is prevalent and well understood–examples of these include virtual networks, storage and blade systems, web server farms, application server clusters, database clusters, and many more. However, the virtualization of management of IT resources is relatively unprecedented.

The idea behind virtualization of management is to take various management capabilities (such as provisioning and configuration, performance and availability monitoring) that are typically well understood when applied to individual resources, and apply them to a new virtual but clearly identifiable and addressable entity called an IT service. Figure 3-2 below shows the relationship between IT resources, IT services, and the management capabilities available for IT services. The figure also includes the typical lifecycle of an IT service.

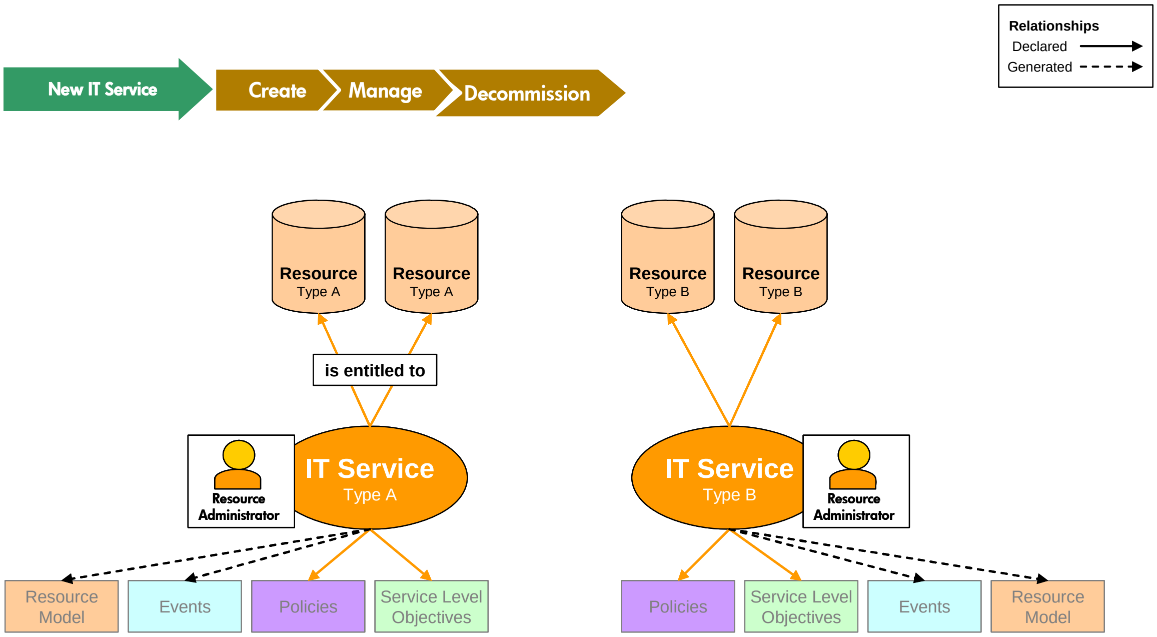

Figure 3-2: Model View of an IT Service

The virtualization of management capabilities is governed by a set of user configurable policies. Performance and availability of the underlying resources may be specified on a resource contained in an IT service using a set of SLOs (see Chapter 2). At runtime, any violations of these SLOs may generate management events that may be used for human or machine consumption.

The management capabilities of IT services are offered externally using a set of Web services. These Web service interfaces are documented in the *SOA Manager Integration Guide* and can be published in a UDDI registry. Opening up the management interfaces in an open and standards, compliant manner provides the fundamental ability to use SOA Manager to create integrated management solutions. Integration is discussed in the following chapter.

IT Service Types

While this section described the concept of an IT service in the abstract, the current version of the SOA Manager's service model implements and understands five types of IT services:

WS Container Services

This type of IT service captures the management of WS containers and their hosted Web services. The WS Container IT service supports the deployment, discovery, and SLO monitoring of a Web service implementation deployed to multiple Web services containers. A WS Container must expose its manageability using a WSM Agent (J2EE or .NET).

WS Intermediary Services

This type of IT service captures the management of WS Intermediaries and their hosted brokered services. The WS Intermediaries IT service supports the deployment, discovery, and SLO monitoring of a brokered service. The WSM Broker is a WS intermediary and is the only intermediary currently supported in the SOA Manager.

Database Services

This type of IT service is used to capture the management of databases that are used by a service-based application. You can also create a database IT service for the SOA Manager's database.

Host Services

This type of IT service is used to capture the management capabilities of a host that is part of a grid that is built using the Globus Toolkit (a de-facto open source grid middleware). The manageability characteristics of a host are accessed using Globus's Monitoring and Discovery Service (MDS) component. Moreover, the MDS component utilizes the Ganglia monitoring system as an information provider to capture low-level metrics and attributes of resources such as CPU load or number of processes. Globus Toolkit and Ganglia are not packaged with the SOA Manager product. For more information about the Globus Toolkit and Ganglia, refer to Globus Toolkit (version 4) and Ganglia Monitoring System. Grid-related offerings from HP can be found at http://www.hp.com/go/grid.

Grid computing began in high-performance technical computing as a way to share widespread computing resources. In enterprise IT environments, grid computing is now gaining adoption rapidly. We define a grid as the software environment for sharing loosely-coupled infrastructure and services. This environment is enabled by grid middleware, which optimizes the utilization of the computing resources it manages. Grid users, who may span organizational boundaries, access the computing services provided by the grid environment. They need not be aware of the location of the underlying physical resources, since the grid virtualizes them. With the migration from technical computing to enterprise IT setting, grid computing has switched from using ad-hoc protocols to a service-oriented architecture based on web service protocols and industry standards. So the computing services provided by the grid have standard interfaces that are discoverable.

The Ganglia monitoring system provides an extension mechanism to incorporate additional user defined metrics that can be captured by running custom scripts or executables. However, the current SOA Manager release requires a certain subset of the metrics supported natively in Ganglia to be made available to it for tracking. The metrics are listed in the *Administrator Guide*.

MOM Services

This type of IT service is used to capture the management of JMS servers. The JMS server must expose its manageability using the JMS Agent. The current SOA Manager release only supports the JMS server included with WebLogic Server.

- Additional IT services will be supported as required.

IT Service Stakeholders

IT service resources are typically co-located in the Data Center and have some Business Owner that pays for the provisioning and maintenance of these resources to run their Business Applications. Figure 3-3 below shows the relationship between a resource owner, the resource administrators, and an IT service.

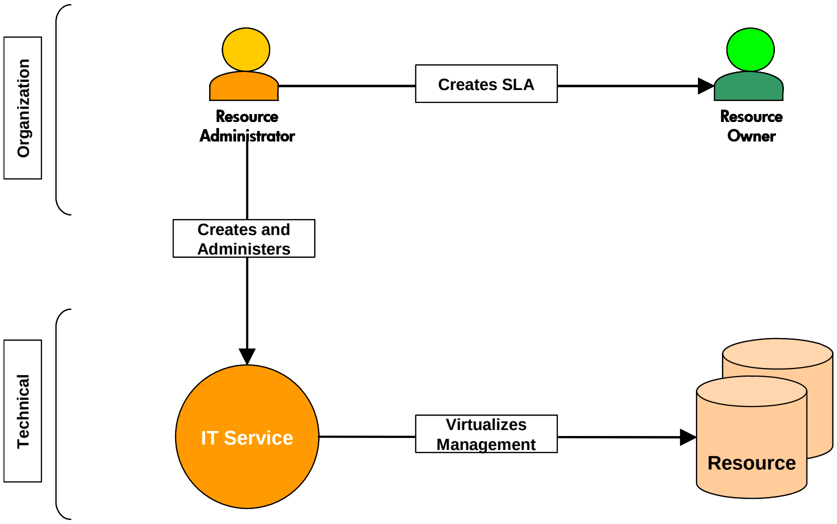

Figure 3-3: Organizational View of an IT Service

Various groups in IT have expertise in these different types of resources and are responsible for the various activities related to managing these IT resources. These activities include: provisioning, deployment, configuration, monitoring, problem management, change management, control, automation, versioning and upgrades. Moreover, each type of resource typically has some IT Operations and Support Contact. The ***Owner*** and ***Support*** contacts are examples of stakeholders or ***People*** that are involved in IT.

Business Service

A ***Business Services*** is the virtualization of some business application that is offered by a business manager to either internal or external customers. Business applications are created and maintained by the IT department and are typically initiated and sponsored by business managers. Business applications are created to meet the needs of internal or external customers and typically represent some business product to the business manager. Figure 3-4 below shows the relationship between a business manager, the IT department, and a business service.

The SOA Manager only models business services representing Web services. Because of this one-to-one relationship, the term business service is often used interchangeably with an offered or consumed Web service.

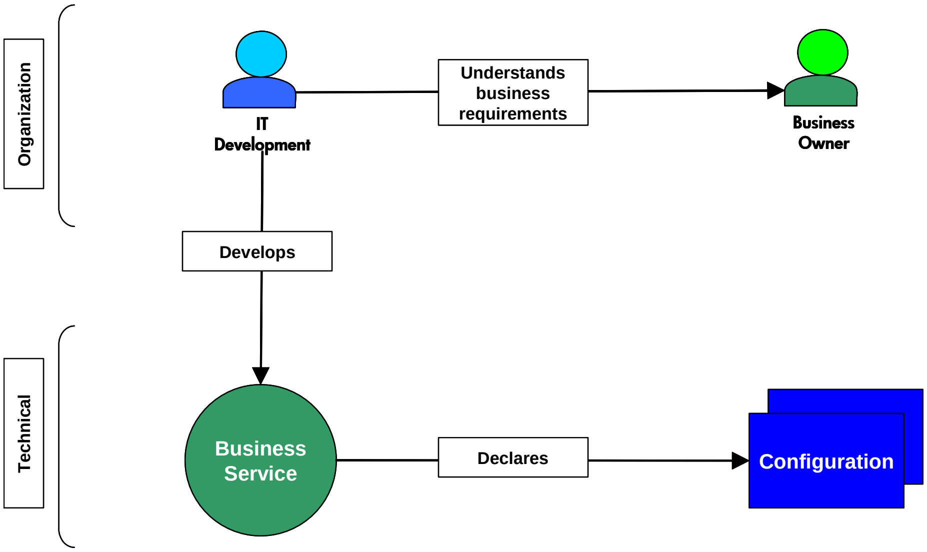

Figure 3-4: Organizational View of a Business Service

Business Service Lifecycle

The SOA Manager’s business service model draws together different groups in an organization and provides an overall means of managing the lifecycle of business services. The model allows various stakeholders to drive a business service’s requirements, definition, development, provisioning, deployment, service level agreement, management, versioning and upgrade.

Business Manager View

Business managers offer a business service to a partner or a customer. As part of the offering, certain SLO requirements (see Chapter 2) are defined. These SLOs define performance and availability objectives for the business service. For example, a business manager may decide to offer an Order Status Query service to customers. The service needs to be available to consumers between certain business hours and return responses within some defined response time. Business managers are concerned with whether the offered business service is meeting the functionality and SLOs. However, they would typically not care about the details of how the business service is being implemented.

IT Development View

IT Development is responsible for creating the underlying resources that are required to deliver the business service. These resources may include Web services, database tables, or Message Oriented Middleware that is used to integrate with legacy backend systems. IT Development includes analysts, architects, and developers.

IT Administration View

IT administration is responsible for configuring, deploying and connecting the business services assets across various IT elements. For example, in order to offer a Web service to its ultimate consumers, various groups in IT may have to deploy the implementation on an Application server (potentially clustered), configure the database correctly, configure WSM intermediaries, set up networking equipment to route to the service, register the service in UDDI, and set up appropriate access rights.

Managing Business Services

Lastly, end-to-end management of a business service requires management integration that ties together the management information of business services and IT services. This work is done by IT Development staff in conjunction with IT Operations staff. Once the service is deployed and operational, application support and IT operations people take over the support and maintenance activity.

Figure 3-5 below shows the business service portion of the service model including the lifecycle, stakeholders, and containing elements. Notice that the service model includes relationships among business services.

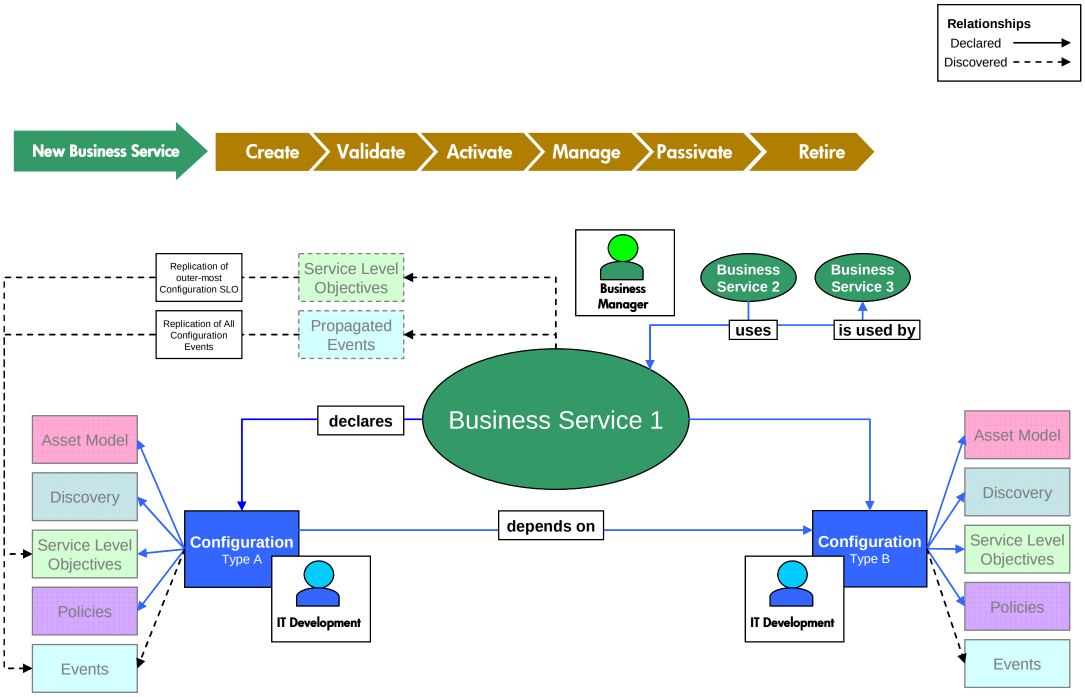

Figure 3-5: Model View of a Business Service

Business Service Configurations

Every business service contains one or more configurations. Business service configurations exist in the service model in order to link business services with the IT services that contain managed resources (e.g., Web services). Once an IT service is linked to a configuration, the managed resources can be added to the configuration and are ultimately managed through the business service.

Configurations are specific for each IT service type. The configuration types include:

- WS Container configurations
- WS Intermediary configurations
- Database Services configurations
- MOM Services configurations
- Host Services configurations

The use of configurations also allows the SOA Manager to provide automation features such as automatic resource discovery, automatic resource deployment, and automatic endpoint routing.

Defining Service Models

The BSE provides a graphical way of creating, editing, and viewing service models. The service model functionality is spread across different screens; each screen is specific for the structural element of the model being defined.

➤ In addition to the BSE, much of the service model can be created, edited and viewed by using integration interfaces. Integration is discussed in the next chapter. This section only discusses the use of the BSE.

The following tasks outline the typical manner in which a service model is defined. This section does not provide detailed procedural steps. Detailed procedures for these and many other tasks are included in the *SOA Manager Administrator Guide*.

In general the steps for creating the service model include:

- **Create an IT Service** – These steps are often performed by IT administrators or operators who are aware of the IT services that are required to deliver a business application.
- **Create a Business Service** – These steps are often completed by a business manager who is responsible for providing a business application to internal or external customers.
- **Add a configuration** – These steps are often completed by IT Development members who are aware of the IT services that are required to deliver a business service.
- **Add Managed Resources** – These steps are often completed by IT Development members who are aware of the resources (e.g., Web services) that are required to deliver a business service.

- **Set SLO Policies** – These steps are often completed by IT administrators or operators who are responsible for monitoring the health of the resources. However, SLOs are typically defined by business managers.

4

Enterprise Management Integration

This chapter provides conceptual information about integrating with the SOA Manager. The information includes:

- **Overview**: This section provides an overview of enterprise management integration.
- **General Integrations**: This section provides a description of the integration features associated with general integrations.
- **Database Integrations**: This section provides a description of the integration features associated with database integrations.
- **Java API Integrations**: This section provides a description of the integration features associated with Java API integrations.
- **Web Service Integrations**: This section provides a description of the integration features associated with Web Service integrations.
- **OpenView Integrations**: This section describes integrations between the SOA Manager and other OpenView Products.

In general, integrators should be familiar with the SOA Manager software before attempting any integration. See the *Integrator Guide* for detailed integration instructions.

Overview

Enterprise Management Integration is the ability to leverage and/or customize the SOA Manager in order to create custom management solutions. Most enterprises have current investments in tools and management products that must be leveraged together with the SOA Manager. Moreover, many enterprises have unique management requirements that the SOA Manager must be able to address. The SOA Manager's integration architecture and features are designed to allow for both maximum reuse and customization.

➤ Integration is not a requirement when using the SOA Manager. That is, the SOA Manager is a standalone solution.

Integration Points are areas of the SOA Manager where integration is possible. One or more integration points may be used depending on the type of integration being performed. The most common integration points are listed below and discussed throughout this chapter.

- **General** – These integration points are used to manually create SOA Manager assets as well as customize the Network Services server and the WSM Broker Agent startup.
- **Database** – This integration point is used to create custom audit reports or to provide audit information within other management applications.
- **Java API** – This integration point is used to create custom management handlers that address an organization's specific Web service management needs.
- **Web Services Interfaces** – This integration point is used to create new (or augment existing) enterprise management applications by reusing the SOA Manager's management data and management model. The data and model are exposed as management Web services.

Lastly, the SOA Manager integrates with some OpenView products. Some of these integrations provide additional functionality to the SOA Manager; while, some integrations allow the SOA Manager's management data to be shared by other OpenView products.

General Integrations

General integrations provide the following benefits:

- Customize startup behaviors.
- Automate repetitive tasks.
- Save time when updating SOA Manager assets over multiple installations.
- Reuse current tool sets (i.e., IDE, Content Versioning System, etc…).
- Maintain current development processes.

Startup Tasks

The Network Services server and WSM Broker Agent are server processes that have the ability to execute user-defined Java code at startup. This is useful for initializing any required services when the processes are started or executing any type of process initialization code.

Integrators are responsible for creating the user-defined Java code as well as configuring each server to use the custom code. Configuration of startup classes is done in the servers' XML configuration file (`server.xml`).

Brokered Services

The WSM Broker’s Broker Configurator is typically used to create brokered services, configure a brokered service's management handlers, and deploy brokered services. These steps can become repetitive and time consuming depending on the number of brokered services that are being deployed. However, these tasks can be completed without using the Broker Configurator.

Integrators are responsible for creating the brokered service definition file (a proprietary file written using XML), packaging the brokered service as a Java Archive (jar file), and copying it to the appropriate directory on the WSM Broker Agent host. Depending on the requirements, some or all of these tasks can be automated.

Java API Integrations

Policy handlers implement management logic that is used to manage Web services. These handlers are inserted in the HTTP or SOAP pipeline that is responsible for processing request and response messages. Brokered services use a set of standard handlers and can also be configured to use a set of simple or advanced handlers. However, an organization may have special management requirements that are not covered by any of the provided handlers.

In such cases, the SOA Manager’s Java API can be used to create custom handlers that can implement any management or processing logic that is required. See “Enabling Manageability” in Chapter 2 for more information about the management architecture and policy handlers.

Integrators are responsible for:

- Creating the custom handler Java logic by extending a base class
- Compiling the handler
- Configuring the handler in the brokered service definition file
- Packaging the handler in the broker service jar file

Security Customization

The SOA Manager’s Java API integration point is used is to create custom security implementations that allow an organization to enforce security policies that are not covered by the WSM Broker’s default security features. This includes custom authentication based on user profile, custom security handlers, and an XML Introspection service.

Database Integrations

Web services audit trace messages are persisted to a central database. This information in conjunction with the model relationships which are also populated into the database (such as which Business Service contains which Web services) can be used to enable reporting and analytical applications such as SLA reporting, billing, non-repudiation, forecasting, etc. Any application can potentially connect to the SOA Manager database and make use of the data. For example, the database can be used to create audit reports using packages like Crystal Reports.

Integrators are responsible for creating and maintaining database connections from within their applications. In addition, integrators are responsible for upgrading their applications as the SOA Manager's database schema changes.

Web Services Integrations

The most common and robust method for integrating with the SOA Manager is by using the SOA Manager's published Web services interfaces. The Web services are SOAP-based Web services that are defined using WSDL. The Web services follow standard Web services management protocols.

Web services integrations allow system architects to leverage current management investments and provide a broader and more thorough view of the enterprise. In general, Web Services integrations are used to:

- Link the SOA Manager with other enterprise management products to create composite and custom applications
- Create and/or reuse current management consoles to display the SOA Manager's management data
- Create custom management consoles

Northbound Interfaces

Northbound interfaces are used to interact with the SOA Manager's management model. The northbound interfaces include the entire Management Information Model. The interfaces include operations to:

- Create business services
- List business services
- List IT services
- Get metrics, status, and alerts
- Query audit traces
- Dismiss alerts
- Delete business services
- Send and view events
- View SLO alerts

The ***Management Information Model*** is a set of consumable Web services (based on various WS standards) and is discoverable through meta-data populated in a UDDI registry. As the underlying standards evolve, the bulk of this model description should remain unaffected.

The HP SOA Manager product provides the starting point framework for populating this information model, and the model is expected to grow over time as HP adds more concrete participants to this model and more importantly, as other partners and vendors contributing to the model extend this in different areas. The model remains purely informational, such that it can be mapped into various technical implementations.

Southbound (i.e. Consumed) Interfaces

The Southbound interfaces are used to interact with a managed WS Container/WSM Broker. The interfaces include operations to:

- Deploy Web services including configuration
- List Web Services
- Get Metrics
- Get Log Messages
- Reset Metrics
- Change Log Levels
- Get Business Content Alerts
- Get Audit Traces

The southbound interfaces contain both standard WS management interfaces and SOA Manager-specific WS interfaces.

Contact the developer alliance team to get detailed documentation about the southbound interfaces.

SOA Services Model Mapping to UDDI

As described above, management interfaces are based on Web services. Like any Web services, the SOA Manager's Web services can be published to UDDI based on mapping recommendations and used by other management applications.

The detailed mapping of the management information model to UDDI is described in the *Integrator's Guide*.

Role of UDDI in the SOA

SOA implementations use UDDI as a system of record. The positioning of the role of registry technology (such as UDDI) in an SOA has evolved over the past few years since its inception. Originally, the registry was conceived as a central discovery point for design-time and run-time reuse.

However, most recent thinking in this direction is that if the registry is used as the only way to offer and discover Web services in the SOA, it provides an excellent control point to achieve Governance during various stages of development, deployment, and runtime management. Business stakeholders and enterprise architects define various policies that must be adhered to in the enterprise SOA. These policies are captured and attached to various entities in UDDI.

The UDDI Registry is used to achieve the following:

- **Reuse** - capture meta-data about Web services as well as other technology assets so that effective search capabilities from various environments may be written against the registry.

- **Policy Definition** – capture various policy definitions that provide the ability for a business person or enterprise architect to mandate policies on various entities.
- **Capture Services Model** – capture various entities participating in an SOA and their relationships.
- **Integration** – reuse the SOA manager’s service model and management data within other domains. The registry provides a central store for this information.

OpenView Integrations

The SOA Manager provides several integrations with other OpenView products. These integrations not only add functionality to the SOA Manager, they also allow greater visibility into an SOA environment using traditional OpenView products that are deployed in the enterprise. The SOA Manager provides the following OpenView integrations.

Select Access

HP OpenView Select Access provides an identity management solution for securing access to IT services and resources. Among other things, the solution includes security authentication, authorization, and administration.

Select Access is typically used to:

- Replace the default security provider that controls access to the BSE and the Broker Configurator Web applications. This allows for single sign-on scenarios where policies for user authentication are pre-established in the enterprise.
- Provide authentication and authorization for the management communication channel between the Network Services server, WSM Agents, and WSM Brokers.
- Provide authentication and authorization for consumers of Web services that are being managed by the WSM Broker Agent.

For information on installing the Select Access integration components, see the *SOA Manager Administrator Guide* located in the `/Documentation` directory of the distribution.

OVIS

OVIS provides a single integrated view of Internet and related services. It is designed to help IT staff efficiently predict, isolate, diagnose and troubleshoot problems, anticipate capacity shortfalls, and manage and report on service level agreements.

The SOA Manager OVIS probe imports business service performance data into OVIS, measuring availability, response time, and other performance metrics. This integration provides greater visibility of the overall health of your business services by leveraging the advanced SLO monitoring, SLA conformance, trend analysis and reporting provided by OVIS.

For information on installing the SOA Manager OVIS probe on the OVIS Management Server, see the OVIS Integration Instructions located in the `/Documentation` directory of the distribution.

OVO

The HP OpenView Operations for UNIX (OVO-U) software provides a fully integrated management solution for networks, systems, databases, and applications found in heterogeneous distributed IT environments. This comprehensive product suite represents a complete set of tools enabling IT organizations to improve overall availability and reliability, maintain the highest degree of management flexibility, and establish management control over virtually all aspects of an enterprise environment.

The SOA Manager OVO-U integration gives OVO-U the ability to show a graphical representation of the service model and view/acknowledge the SOA Manager's alerts. This information can be managed together with other enterprise management data. The integration provides a means of improving the availability of enterprise resources as well as a comprehensive and centralized view of the health and well being of the resources.

For information on installing the SOA Manager OVO-U Plug-in on the OVO Management Server, see the *HP OpenView Service Oriented Architecture Manager OVO/Unix Plug-in User Guide* located in the `/Documentation` directory of the distribution.

5

Use Cases

This chapter presents some common use cases that users of the SOA Manager software typically perform. The sections in this chapter include:

- **Overview**: This section provides a basic overview of the use cases.
- **Installation and Setup**: This section provides the tasks that are associated with installing and setting up the different SOA Manager components.
- **Configure Manageability**: This section provides the tasks that are associated with configuring Web services management policies.
- **Define Services Model**: This section provides the tasks that are associated with managing SOA resources using the service model concept.
- **Monitor Availability**: This section provides the tasks that are associated with monitoring SOA resources availability at runtime.
- **Monitor Service Performance**: This section provides the tasks that are associated with monitoring Web services performance at runtime.
- **Service Lifecycle**: This section provides the tasks that are associated with managing Web services from a lifecycle perspective.
- **Define Security Policies**: This section provides the tasks that are associated with security.
- **Publish to UDDI**: This section provides the tasks that are associated with publishing application and management Web services to a UDDI registry.

Overview

The use cases in this chapter provide a general concept of how to use the SOA Manager software. As a general rule, the use cases are listed sequentially and suggest a logical method of moving through the software. The list of use cases is not exhaustive and should only be used as a starting point.

The use cases are presented as if a single user is completing the tasks. However, it is more likely that many individuals within an organization will work to complete a use case. These individuals include developers, administrators, operators, support, and business managers.

Lastly, each use case is split into one or more tasks. The tasks summarize how to complete the use case. Some tasks are only completed once. Other tasks are completed as part of normal day-to-day operations. Tasks that are relevant in multiple use cases are repeated in each use case but may only have to be completed once. The tasks are mainly informative. More detailed procedures are included in the documentation located in `/Documentation` directory of the distribution.

Installation and Setup

The tasks associated with the installation and setup use case vary depending on an organization's Web services deployment architecture and management requirements. However, a minimum installation and setup involves the following components:

- A single Network Services Server
- One or more management agents (WSM Broker and/or WSM Agents)

This section provides a high-level overview of the tasks that are needed to install and set up these components. This section does not provide detailed procedural steps. Detailed procedures for these and many other tasks are included in the administrator guides located in the `/Documentation` directory of the SOA Manager distribution.

Network Services Server

The Network services server is installed using the installer executable included with the SOA Manager distribution. An installer is provided for Windows, HP-UX, and Linux. For installations on Windows, the Network Services server can be installed as a Windows Service.

There are several setup tasks that need to be completed after the installation is completed. The tasks are listed below and are completed using various configuration files or the BSE console.

- **Set up database connectivity for Network Services persistence**

  The HSQL database is included with the SOA Manager software and is preconfigured out-of-box. The database is used to simplify testing and evaluation scenarios. However, for production environments, schemas are provided to create the database tables for Oracle. The Network Services must be configured to use a database other than the HSQL database. Database configuration is completed in the servers configuration file (`mipServer.xml`).

- **Set up UDDI connections**

  Management Web Services (specifically those that expose parts of a service model) can be published to a UDDI Registry for reuse by other applications. To enable this functionality, the Network Service server must be configured to connect to the UDDI Registry. UDDI configuration is completed using the BSE.

- **Configure PKI infrastructure by installing appropriate X.509 certificates into a Java Key Store**

  Management communication between the Network Services server, WSM Brokers, and WSM Agents can be secured using HTTPS and SSL. X.509 certificates for the WSM Broker and/or WSM Agents must be added to the Java Key Store used by the Network Services server. To enable this functionality, the Network Services server must be configured to connect to the Java Key Store. Security settings, including Java Key Stores, are configured in `mipServer.xml`.

- **Configure connection to Security Policy Decision Point**

  The Network Services server has the ability to connect to HP Select Access, which provides a single control point for authentication, authorization and identity administration. To enable this functionality, a Select Access add-on component (located in the distribution) must be installed on the Network Services server. The component communicates with the Select Access Administration server. In addition, two files must be configured: `mipServer.xml` and `selectaccess.properties`.

- **Configure e-mail server (SMTP)**

  Alerts generated by network services can be sent directly to the e-mail address of a stakeholder or indirectly via alert categories. E-mail server settings are configured in the BSE console.

- **Configure alert targets and alert categories**

  Alerts that are generated by the Network Services server can be sent to targets such as e-mail, log files, and SNMP. Multiple alert targets can be grouped together into alert categories. Alerts that are sent to an alert category are automatically sent to all targets in the category. Alert targets and alert categories are defined using the BSE console.

- **Configure stakeholders**

  Most resources that are part of the service model can be associated with an owner and support person. Before you can associate these people with a resource, they must be added to the Network Services server. The information includes the person's name and e-mail address. Stakeholders are defined and associated to a resource using the BSE console.

WSM Broker

The WSM Broker is installed using the installer executable included with the SOA Manager distribution. An installer is provided for Windows, HP-UX, and Linux. For installations on Windows, the WSM Broker can be installed as a Windows Service.

➤ The Network Services server and the WSM Broker are installed using the same installer. At this time, the Network Services server and the WSM Broker cannot be installed separately.

There are several setup tasks that need to be completed after the installation is completed. The tasks are listed below and are completed using configuration files and the Broker Configurator console.

- **Configure PKI infrastructure by installing appropriate X.509 certificates into a Java Key Store**

  Application communication between SOAP clients, the WSM Broker, and Web Services Containers can be secured using HTTPS and SSL. X.509 certificates for the Web Services Containers must be added to the WSM Broker's Java Key Store and the WSM Broker's X.509 certificate must be added to any SOAP clients. To enable this functionality, The WSM Broker must be configured to connect to a Java Key Store. Security settings, including Java Key Stores, are configured in `mipServer.xml` and can also be configured using the Broker Configurator console.

- **Configure connection to Security Policy Decision Point**

  The WSM Broker has the ability to connect to HP Select Access, which provides a single control point for authentication, authorization and identity administration. To enable this functionality, a Select Access add-on component (located in the distribution) must be installed on the WSM Broker server. The component communicates with the Select Access Administration server. In addition, two file must be configured: `mipServer.xml` and `selectaccess.properties`.

WSM .NET Agent

A WSM .NET Agent is installed on every IIS computer that hosts Web services that are to be managed. The Agent is an application that is deployed in IIS. The WSM .NET Agent is installed using an installer executable included with the SOA Manager distribution. All necessary setup is performed by the installer.

WSM J2EE Agent

A WSM J2EE Agent for WebLogic Server (WLS) is installed on every WLS computer (administration server and managed servers — both independent and in clusters) that hosts Web services that are to be managed. The WSM WLS Agent is included with the SOA Manager distribution. The Agent is packaged as a zip file and is installed using a set of scripts. Some configuration of the WLS is required. The agent is itself an application that is deployed in WLS.

Configure Manageability

The configure manageability use case allows you to enable/disable different management policies for Web services. Management policies are implemented by the management agents and are configured differently depending on the agent you are using.

The agents use a set of default management policies (e.g., for collecting performance data). Additional management policies (e.g., auditing) must be configured as needed.

WSM Broker

The Broker Configurator console is used to configure manageability for a Web service. Two steps are required:

- **Create brokered services**

  The WSM Broker hosts Brokered Services, which are Web service proxies to the Web services that are to be managed. A brokered service is created for each Web service that is to be managed. Brokered services can be created for SOAP-based Web services and XML-based Web services. In addition, brokered services can be either simple or custom depending on the type of manageability that is required.

  All application requests must be sent to the brokered service in order for management data to be collected. The WSM Broker is responsible for dispatching the request to the actual service endpoint.

- **Enable management policies**

  Management policies must be configured separately for each brokered service. Management policies can be configured when a brokered service is created, or the brokered service can be edited at any time to enable/disable management policies. Some of the management policies include: security, auditing, business content alerts, and logging. Different management policies are available depending on the brokered service type (simple or custom).

WSM .Net Agent

The .NET Agent uses SOAP extensions to add manageability to a Web service. Default SOAP extensions are applied to a managed Web service when the service is discovered by the Agent. However, the SOAP extensions for auditing, business metric alerts, and logging are not enabled by default. These SOAP extensions are configured in either a .NET application's `Web.config` file or in the .NET server's `Machine.config` file. These files are both XML-based files.

WSM J2EE Agent

The J2EE Agent contains a set of common handlers (JAX RPC Handlers) that are used to provide management for Web services. There are three handlers: Monitoring handler, Auditing handler, Business Metric handler.

Common handlers are enabled and disabled in a Web services' `web-services.xml` deployment descriptor. By default, the Monitoring and Auditing handler are added to this file when a Web service is discovered and therefore are enabled. However, you must manually add the Business Metric handler to enable business content monitoring.

Define Services Model

The service model definition use case is used to define a model that represents the SOA resources that are being managed. These resources cannot be viewed unless they are part of a service model. The model is defined using the BSE console, which is also used to view the management model and actively monitor the resources that are part of the model. The following tasks are used to define a service model for managing SOA resources.

- **Define IT Services and associate them with resources**

  Resources that are to be managed must be contained within an IT service. IT services are created for specific types of resources. For example, a WS container IT service would be created for managed WS containers. As part of the IT service definition process, an IT service is assigned an owner and support contact.

  After an IT service is created, one or more resources can be added to the IT service. For example, a WS container IT service can contain any number of managed WS containers. Because the WS container has an installed instance of the WSM agent, the Web services in the container are automatically discovered when the container is added to the IT service.

- **Define Business Services and their Configurations**

  Business services are the main context of the service model and the main context by which all resources are viewed in the BSE. This task is completed in two steps:

  — Create the business service – In this step, an empty business service is created.

  — Link an IT Service — In this step, multiple IT service configurations can be created, by linking existing IT Services. An appropriate IT Service configuration is created along with the link to the appropriate resource. This step can also be accomplished by following the next two manual steps in place of this step.

    — Add a Configuration – In the first step, one or more business service configurations are created for the business service. Different configurations can be created for each IT service type. For example, a WS container configuration would be created for resources that are part of a WS container IT service. As part of this step, the configuration is bound to an IT service and is assigned an owner and support contact.

    — Add a Resource – In the second step, any number of application resources are added to a configuration. Because the configuration is bound to an IT service, any resource within the IT service can be added to the configuration. For example, any Web services resources that are associated with a WS container IT service can be added to a WS container configuration. As part of this step, a resource is assigned an owner and support contact.

- **Define Service Level Objectives (SLO) and tie their breach notifications to Alert categories**

  This task is used to define acceptable operating limits for managed Web services. In this task, threshold values are defined for a Web service's performance metrics. For example, a Web service resource that is contained in the service model is edited and a threshold value of 100 Milliseconds is defined for the maximum response time. A response time greater than 100 Milliseconds results in an alert being generated. As part of the SLO definition, specific alert categories can be assigned to receive SLO breach alerts.

- **Define Relationships between Business Services**

  A business service may use, or be used by, any number of other business services. This relationship has to be known and declared in the service model and represents a dependency relationship between the Web services in one business service to that in another. This dependency relationship is used for impact analysis and root cause analysis.

Monitor Availability

The monitoring availability use case allows key stakeholders to be notified whenever a business service, IT service, IT service resource, or Web service becomes available or unavailable. Monitoring availability is often used as a first step when troubleshooting problems.

The BSE is used to enable or disable availability alerts and is also used to view availability alerts. Moreover, any alert target that is set up in the Network Services server can receive an availability alert. The following tasks are used to perform availability monitoring. Some of these tasks may have been completed as part of another use case, but are included here for thoroughness.

- **Configure alerts targets and alerts categories**

  Availability notifications can be sent to targets such as e-mail recipients, log files, and SNMP. Multiple alert targets can be grouped together in alert categories. Alerts that are sent to an alert category are automatically sent to all targets in the category. Alert targets and alert categories are defined using the BSE console.

- **Enable availability notifications**

  Availability notifications are enabled and disabled separately for each business service, IT service, IT service resource, and Web service using their respective edit screens. The edit screen also contains an Alert Recipients section that allows you to select an alert category to which availability notifications are sent.

- **Receive availability notifications**

  When an availability notification is generated, it is viewable in the BSE and sent to any configured alert targets. The notification includes the name of the business service, IT service, IT service resource, or Web service and a simple message that indicates whether it is available or unavailable.

Monitor Service Performance

The monitor service performance use case allows application administrators or key stakeholders to see how well a Web service is performing and how its performance affects an overall business service that may be comprised of many Web services.

The term performance is used here to represent the general health of a service. Performance is computed by comparing various performance metrics (e.g., average and maximum response times, number of faults, total number of messages, etc…). The metrics are viewed over time and provide an opportunity to see performance trends.

Lastly, performance metrics are compared against pre-defined SLO warning and SLO breach thresholds. A stakeholder can receive e-mail notifications when SLO violations occur. These alerts are accompanied by console messages that an operator can review. Console messages appear in the BSE.

The following tasks are used to monitor and troubleshoot service performance. Some of these tasks may have been completed as part of another use case, but are included here for thoroughness.

- **Configure alerts targets and alerts categories**

  Alerts that are generated by the Network Services server can be sent to targets such as e-mail, log files, and SNMP. Multiple alert targets can be grouped together in alert categories. Alerts that are sent to an alert category are automatically sent to all targets in the category. Alert targets and alert categories are defined using the BSE console.

- **Define Service Level Objectives (SLO) and tie their breach notifications to Alert categories**

  This task is used to define acceptable operating limits for managed Web services. In this task, threshold values are defined for a Web services' performance metrics. For example, a Web service resource that is contained in the service model is edited and a threshold value of 100 milliseconds is defined for the maximum response time. A response time greater than 100 milliseconds results in an alert being generated. As part of the SLO definition, specific alert categories can be assigned to receive SLO breach alerts.

- **View the performance screen**

  Performance metrics are displayed on the Web service performance screen and can also be viewed for specific Web service operations. The screen is used to view all collected metrics as well as see the metrics displayed in a performance graph.

- **Receive SLO alerts and check status**

  When an SLO alert is generated, it is viewable in the BSE and sent to any configured alert targets. The alert includes the name of the Web service, the performance metric value, and the assigned SLO threshold value that was exceeded. In addition, status fields throughout the BSE visually indicate when a Web service has any active alerts. Whenever a performance metric returns within the acceptable SLO threshold value, the status of the Web service returns to normal.

- **View Performance Reports**

  This task is used to view reports in the BSE that contain performance data such as request count, success count, failure count, availability percentage, maximum response time, and minimum response time. The reports can be constrained using a set of query criteria. The criteria can include values such as Web service name and date and time range. The data for the performance reports are retrieved from the SOA Manager's database and viewed on the Reports screen.

Service Lifecycle Management

The lifecycle management use case provides a level of automation and adaptability to the tasks associated with managing Web services throughout their lifecycle. Much of the automation and adaptability comes from the use of the service model that can remain constant as the resources in the model change over time.

The following tasks are used to perform lifecycle management and are completed using the BSE. Some of these tasks may have been completed as part of another use case, but are included here for thoroughness.

- **Define Service Model**

  This task is used to define a service model (see the “Define Services Model” use case). As part of the model definition tasks, IT resources are registered within IT services. All managed Web services and/or brokered services that are contained in an IT resource are automatically discovered and registered within the Network Services server. As new Web services and/or brokered services are added to an IT resource, they are automatically added to the Network Services server.

- **Deploy services**

  This task allows you deploy a Web service or a brokered service to an IT resource (WS container or WS intermediary respectively) that is registered within an IT service. Such assets are deployed to remote IT resources from within the BSE without having to know the deployment technology of a particular IT resource. Once deployed, the assets are automatically added to the Network Services server.

- **Discover services**

  This task allows you to configure a service model to use a discovery pattern in order to identify a Web service or a brokered service that is to be part of the service model. Once the Web service or brokered service is registered in the Network Services server, it is automatically discovered using the discovery pattern and added to the service model. This allows the model definition to remain constant even though the underlying assets may not be deployed or are changing and moving between different hosting environments.

- **Configure and monitor services**

  This task allows you to configure SLO definitions for Web services and then monitor the Web services’ performance during runtime. See the “Monitoring Service Performance” use case for complete details.

Define Security Policy

The security use case varies depending on the level of security that is required by an organization. Therefore, some tasks may not be required depending on the type of security that is implemented. This section covers all security tasks.

The tasks needed to implement the security use case are relevant to both the Network Service server and the WSM Broker. When using the WSM Agents, the platform’s (WLS and .NET) security implementation should be used.

Some of the tasks in this section may have been completed as part of another use case, but are included here for thoroughness.

Network Services Server

The following tasks are used to implement varying levels of security when using the Network Services server. Detailed instructions for implementing security can be found in the *SOA Manager Administrator Guide*.

- **Configure PKI infrastructure by installing appropriate X.509 certificates into a Java Key Store**

Management communication between the Network Services server, WSM Brokers, and WSM Agents can be secured using HTTPS and SSL. X.509 certificates for the WSM Broker and/or WSM Agents must be added to the Java Key Store used by the Network Services server. To enable this functionality, the Network Services server must be configured to connect to the Java Key Store. Security settings, including Java Key Stores, are configured in `mipServer.xml`.

- **Configure connection to Security Policy Decision Point**

  The Network Services server has the ability to connect to HP Select Access, which provides a single control point for authentication, authorization and identity administration. To enable this functionality, a Select Access add-on component (located in the distribution) must be installed on the Network Services server. The component communicates with the Select Access Administration server. In addition, two files must be configured: `mipServer.xml` and `selectaccess.properties`.

- **Secure the management channel**

  This task is used to configure the Network Services server to use HTTPS and SSL when communicating management information between itself and any WSM agents. In order to secure the management channel, each WSM Agent must first be configured to use a secure port. Management channel security is enabled when registering an IT resource in an IT service.

- **Secure BSE login**

  This task is used to secure access to the BSE console. This can be done using both HTTPS /SSL and Select Access. When using HTTPS and SSL, a secure HTTPS port must be configured in `mipServer.xml`. In addition, any browser used to access the BSE must contain a Certificate Authority (CA) root certificate from the CA that was used to verify the Network Services server's SSL certificate.

  When using Select Access, you must create a Select Access resource for the BSE using the HP OpenView Select Access Policy Builder.

WSM Broker

The following tasks are used to implement varying levels of security when using the WSM Broker. Detailed instructions for implementing security can be found in the *SOA Manager Administrator Guide* and the *WSM Broker Administrator Guide*. Many of these tasks are similar to the tasks for the Network Services server. However, because the Broker receives application traffic, some additional security precautions can be implemented.

- **Configure PKI infrastructure by installing appropriate X.509 certificates into a Java Key Store**

  Application traffic between SOAP clients, the WSM Broker, and Web Services Containers can be secured using HTTPS and SSL. X.509 certificates for the Web Services Containers must be added to the WSM Broker's Java Key Store and the WSM Broker's X.509 certificate must be added to any SOAP clients. To enable this functionality, The WSM Broker must be configured to connect to a Java Key Store. Security settings, including Java Key Stores, are configured in `mipServer.xml` and can also be configured using the Broker Configurator console.

- **Configure connection to Security Policy Decision Point**

The WSM Broker has the ability to connect to HP Select Access, which provides a single control point for authentication, authorization and identity administration. To enable this functionality, a Select Access add-on component (located in the distribution) must be installed on the WSM Broker server. The component communicates with the Select Access Administration server. In addition, two files must be configured: `mipServer.xml` and `selectaccess.properties`.

- **Secure the management channel**

  This task is used to secure management communication between the Broker and the Network Services server. Communication can be secured using both SSL and Select Access. Both are enabled in `mipServer.xml`.

- **Secure the application channel**

  This task is used to secure communication between a consumer, the Broker, and a WS Container. Communication on this channel can be secured at both the transport layer (SSL and HTTPS) and at the message layer (WS-Security). Both layers can utilize Select Access for authentication and authorization.

  Application channel security is enabled using security policies. Security policies are configured using the Broker Configurator and are configured per brokered service. When using Select Access, you must also create Select Access resources for the brokered services using the HP OpenView Select Access Policy Builder.

- **Secure Broker Configurator login**

  This task is used to secure access to the Broker Configurator console. This can be done using both SSL and Select Access. When using SSL, a secure HTTPS port must be configured in `mipServer.xml`. In addition, any browser used to access the Broker Configurator must contain a CA root certificate from the CA that was used to verify the Broker server's SSL certificate.

  When using Select Access, you must create a Select Access resource for the Broker Configurator using the HP OpenView Select Access Policy Builder.

Publish to UDDI Registry

The UDDI use case is used to take advantage of a UDDI registry in order to publish, discover, and use Web services. As discussed in the previous chapter, if the UDDI registry is used as the sole means by which Web services are discovered, then it also provides an excellent control point to achieve Governance during various stages of development, deployment, and runtime management.

There are two types of Web services that are published from the SOA Manager to a UDDI registry: application Web services and management Web services. Application Web services are the services which are being managed. Management Web services are services that are used to expose and interact with the management service model.

The following tasks are used to publish Web services to a UDDI registry. Some of these tasks may have been completed as part of another use case, but are included here for thoroughness.

- **Set up UDDI connections**

  This task is used to configure the Network Service server to connect to a UDDI registry. UDDI configuration is completed using the BSE Settings screen.

- **Publish business services**

  This task is used to publish the WSDL for the business service management web service, the WSDL for the Web service that is managed by the business service, relationships between business services, and organizational contact information. Business services are published to UDDI using the BSE Publish screen that can be accessed from a Business Service's View screen.

- **Publish IT services**

  This task is used to publish the WSDL for the IT service management web service and organizational contact information. IT services are published to UDDI using the BSE Publish screen that can be accessed from an IT service's View screen.

Glossary

Application Channel

Application channel refers to the request/response communication between an application client, such as a browser, and an application component such as a Web service.

Auditing

Auditing is a management feature that captures trace information for all Web service requests and responses.

Availability Monitoring

Availability monitoring is a management feature that is used to monitor the availability of SOA resources such as Web services.

Broker Configurator

The Broker Configurator is the WSM Broker's administration console. It is used to create and configure brokered services as well as configure the Broker's server properties.

Brokered Services

A brokered service is a proxy to a final Web service endpoint and is used to enable the management of a Web service.

Business Services

A business service is the virtualization of some business application that is offered by a business manager to either internal or external customers.

Business Service Configuration

A business service configuration is a part of the service model that contains an IT service and provides varying levels of automation.

Business Service Explorer (BSE)

The BSE is the SOA Manager's management console. It is used to create service models and monitor SOA resources.

Content Monitoring

Content monitoring is a management feature that searches Web service request and/or response messages for specific content.

Distributed Management

Distributed management is an approach to managing resources that are deployed and distributed across an enterprise network environment.

Enterprise Management Integration

Enterprise management integration is the ability to leverage and/or customize the SOA Manager in order to create custom management solutions.

Grid

Grid computing began in high-performance technical computing as a way to share widespread computing resources. In enterprise IT environments, grid computing is now gaining adoption rapidly. In this documentation a grid is defined as the software environment for sharing loosely-coupled infrastructure and services. SOA Manager manages grid hosts that are constructed from the Globus Toolkit.

Impact Analysis

Impact analysis is the ability to discover how the performance of a service affects other related services.

Integration Points

Integration points provide the ability to either extract information from the SOA Manager or add additional management data to the SOA Manager.

Interposed Manageability

Interposed manageability means inserting management policies in the request/response path of Web services.

IT Service

An IT Service, as defined in the SOA Manager, represents the virtualization of management information or capabilities of a group of resources of a certain type that are associated with a set of stakeholders. An IT service can represent a single IT resource or can represent a collection of resources.

Logging

Logging in the SOA Manager captures the local standard output for Web service containers and Web service intermediaries so that the output can be analyzed from a remote central location.

Managed Object (MO)

An MO is a representation of a managed element such as a Web service. An MO can be related to either a logical or physical piece of the IT infrastructure. In the SOA Manager, MOs are exposed as Web services that provide attributes and operations that can be invoked.

Managed Service

A managed service is a Web service which is being managed by the SOA Manager.

Management Agents

Management agents are software components that get installed on a computer and are responsible for performing management tasks.

Management Channel

The Management channel refers to the communication between the Network Services server and one or more management agents. In the SOA manager, the management channel can be different than the application channel.

Management Information Model

The management information model is a set of Web services (based on various standards such as WSDL, WSDM, etc.) consumable on the wire, and discoverable through meta-data populated in a UDDI registry.

Management Policies

Management policies contain the management logic that is used to interpose visibility and controls on Web services. Management policies are implemented in WSM Agents or the WSM Broker.

Management Proxies

Management proxies are software components that get installed on a computer and are responsible for gathering management data for computers that do not have a native management agent available for them. The WSM Broker is an example of a management proxy.

Management Server

A Management server is a centralized software component that aggregates the data that is gathered by any number of management agents. The Network Services server is an example of a management server.

Management Web Service

A management Web service is a Web service that exposes management information using standard Web services management protocols. The WSM Agents and the WSM Broker expose their management information as management Web services.

Northbound Interfaces

Northbound interfaces are Web services-based integration interfaces that are used to extract the information contained in the SOA Manager's management model.

Performance Monitoring

Performance monitoring is a management feature that captures a set of real-time performance metrics that clearly indicate the health, availability, and performance of Web services.

Policy Handlers

Policy handlers are the actual implementation of the management policies in the WSM Agents and WSM Broker. Policy handlers are often referred to as simply handlers.

Public Key Infrastructure (PKI)

A PKI enables users of a basically unsecure public network such as the Internet to securely and privately exchange data through the use of a public and a private cryptographic key pair that is obtained and shared through a trusted authority. The public key infrastructure provides for a digital certificate that can identify an individual or an organization and directory services that can store and, when necessary, revoke the certificates.

Resource Management

Resource management is the act of managing the SOA resources that are being used by business applications.

Root Cause Analysis

Root cause analysis is the ability to discover which Web service is causing a group of related Web services to degrade.

Secure Sockets Layer (SSL)

SSL is a commonly-used protocol for managing the security of a message transmission over the Internet.

Service

A service is a self contained collection of functionality that promotes a high degree of isolation from internal details while at the same time offering its functionality to other services.

Service Consumer

A service consumer is a participant in a service-based application that uses a service based on the functionality and value that the service provides.

Service Level Agreement (SLA)

An SLA is an agreement between a service consumer and a service provider about the expected level of availability and performance of a service.

Service Level Objective (SLO)

An SLO is a set of preferred operating limits for a Web service.

Service Oriented Architecture (SOA)

An SOA is a set of principles that define an architecture that is loosely coupled and comprised of service providers and service consumers that interact according to a negotiated contract or interface.

Service Model

A service model is the virtual representation of managed SOA resources.

Service Producer

A service producer is a participant in a service-based application that focuses on how the service provides functionality and value and which resources provide the service.

SLO Monitoring

SLO Monitoring is a management feature that evaluates a Web service's performance against an SLO to insure it is within acceptable operating limits.

Simple Object Access Protocol (SOAP)

SOAP is an XML-based protocol that is typically used over HTTP to send messages (commonly referred to as SOAP messages) between application clients and servers. SOAP is the standard for Web services messages and is one of the foundation standards of Web services.

Solution

A set of features and capabilities delivering business value to a customer through a combination of hardware, software, and services.

Southbound Interfaces

The Southbound interfaces are Web services-based integration interfaces that are used to interact with a managed WS Container/WSM Broker.

Trend Analysis

Trend analysis allows operators and administrators to analyze changes in Web service performance over time.

Universal Description, Discovery, and Integration (UDDI)

UDDI is a specification that defines a registry service for Web services that allows Web services to be discovered. UDDI is often referred to as a Yellow Pages of Web services.

Web Service

A Web service is a service that is built using the SOAP and WSDL standards.

Web Services (WS) Container

A WS container represents a SOAP container or environment that can host Web services. AXIS, IIS, and WebLogic Server are examples of WS containers.

Web Service Management (WSM)

WSM is the act of managing the Web services that are being used by business applications. WSM in the SOA Manager software goes beyond managing just Web services to include a range of SOA resources that are equally vital to the success of Web services.

Web Services Management (WSM) Agent

A WSM Agent is an enablement component that is installed in a WS Container in order to manage the Web services in the container as well as the container itself. There is a WSM Agent for both the Microsoft Internet Information Server (IIS) and for the BEA WebLogic Server (WLS).

Web Services Description Language (WSDL)

WSDL is an XML-based language that is used to describe a software component. A WSDL definition describes how to access a Web service and what operations it can perform.

Web Services Distributed Management (WSDM)

WSDM is an OASIS standard that has been formed to define web services management, including using web service architecture and technology to manage distributed resources.

Web Services Intermediary

A Web services intermediary represents a proxy to a WS Container. The WSM Broker is considered a Web service intermediary.

Web Services Management Framework (WSMF)

WSMF is a standard defined by HP and submitted to the OASIS WSDM TC. The standard defines fundamentals for the Management of Web Services (MOWS) and for Management Using Web Services (MUWS).

WSM Broker

The WSM Broker is a flexible, configurable, high performance Java-based Web services intermediary process. The WSM Broker is used to manage Web services that are hosted in containers that do not provide native management for Web services. The WSM Broker is an implementation of a Management Proxy and does not need to be co-located with the Web services being managed.

XML (Extensible Markup Language)

XML is a markup language used to describe data and does not include any presentation logic for the data.

Index