IBM

Power Systems and SOA Synergy

- Explains partitioning and virtualization technologies
- Discusses SOA implementation on System p
- Covers the AIX and Linux platforms

Budi Darmawan
Christian Richter
Gustavo Lima
Mario Ernesto Orozco Salazar
Monty Poppe
Vidyadharan Elambila Veetil

Redbooks

ibm.com/redbooks

International Technical Support Organization

Power Systems and SOA Synergy

June 2008

SG24-7607-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (June 2008)

This edition applies to the following products:
- AIX V6.1
- WebSphere Application Server V6.1
- WebSphere ESB V6.1
- WebSphere Message Broker V6
- DB2 Universal Database Enterprise Server Edition V9.1
- IBM Director V5.20

Contents

Notices vii
Trademarks viii

Preface xi
The team that wrote this book xi
Become a published author xiii
Comments welcome xiv

Chapter 1. Power Systems and SOA synergy 1
1.1 SOA basics 2
1.2 Power Systems and SOA 3

Chapter 2. SOA introduction 5
2.1 Understanding SOA 6
2.1.1 The evolution towards SOA 6
2.1.2 Defining services 8
2.1.3 Defining SOA 12
2.1.4 SOA basics 13
2.2 Benefits of SOA 16
2.3 Common misconceptions 17
2.4 SOA standards, mediation, and registry 19
2.4.1 SOA standards 20
2.4.2 Enterprise Service Bus and SOA 22
2.4.3 Role of the WebSphere Service Registry and Repository 34
2.5 IBM SOA Foundation 39
2.5.1 The SOA life cycle 40
2.5.2 SOA solution stack and reference architecture 48
2.5.3 Logical architecture 54
2.5.4 SOA scenarios 61
2.5.5 SOA entry points 68
2.5.6 Use case to scenario mapping 71

Chapter 3. SOA infrastructure selection 73
3.1 Mapping the infrastructure 74
3.2 SOA requirements on infrastructure 74
3.3 Key elements of a SOA infrastructure 77
3.3.1 Mediation and registry considerations 78
3.3.2 Security 80
3.3.3 Performance 82

3.3.4 Infrastructure management 83
3.3.5 Infrastructure virtualization 85
3.4 Architecting SOA infrastructure 87
3.4.1 Design stages 87
3.4.2 Infrastructural anti-patterns 95
3.5 System p and the infrastructure layer 95
3.5.1 Abstract data access 96
3.5.2 Interacting with external resources 96
3.5.3 Provide new access points 96
3.5.4 Dynamic composition of business environments 96
3.6 Conclusion 97

Chapter 4. IBM System p features 99
4.1 System p and SOA synergy 100
4.1.1 System p architecture 100
4.1.2 SOA and System p architectural synergy 101
4.2 System p feature overview 103
4.2.1 Virtualization on System p 104
4.2.2 Logical partitioning 108
4.2.3 System p availability features 116
4.2.4 System p management features 119
4.3 Operating system features overview 125
4.3.1 Operating system support summary 126
4.3.2 Virtualization features 129
4.3.3 Security features 130
4.3.4 Availability features 140
4.3.5 Manageability features 145
4.4 Realization of System p benefits to SOA 147
4.4.1 Flexible and scalable 148
4.4.2 Highly available and reliable 156
4.4.3 Secure 160
4.4.4 Optimized for reduced costs 161
4.4.5 Simplified management 163

Chapter 5. SOA implementation 167
5.1 Implementation environment 168
5.2 Hardware configuration 169
5.2.1 System plan 170
5.2.2 Virtual servers 172
5.2.3 System plans in Virtualization Manager 175
5.2.4 Virtual I/O Server partition 178
5.3 Software environment 189
5.3.1 WebSphere Application Server V6.1 190

5.3.2 WebSphere MQ V6 . . . 193
5.3.3 DB2 Universal Database Enterprise Server Edition V9.1 . . . 193
5.3.4 WebSphere Message Broker V6.1 . . . 194
5.3.5 WebSphere ESB V6.1 . . . 195
5.3.6 WebSphere Services Registry and Repository V6.1 . . . 196

Chapter 6. Operational environment . . . 197
6.1 ITSO trader example . . . 198
6.1.1 Business view . . . 198
6.1.2 Application view . . . 199
6.2 Using SOA scenarios . . . 201
6.2.1 Gather business requirements . . . 201
6.2.2 Generic use case selection . . . 203
6.2.3 SOA scenario selection . . . 206
6.2.4 Identify, reuse, and apply patterns . . . 206
6.2.5 Leverage implementation guides . . . 206
6.3 The runtime environment . . . 207
6.3.1 Front-end J2EE Web application . . . 208
6.3.2 Back-end implementation . . . 209
6.3.3 Mediation implementation . . . 209
6.3.4 Resources structure of the setup . . . 210
6.4 Application feedback and monitoring . . . 211
6.4.1 IBM Director . . . 211
6.4.2 IBM Tivoli Monitoring . . . 214
6.4.3 Application management tool . . . 217

Appendix A. WebSphere Application Server and WPAR . . . 221
Considerations . . . 222
Creating workload partitions . . . 223
WebSphere Application Server installation wizard . . . 223
Silent installation with profile creation . . . 224
Profile management . . . 225
Update installer . . . 225
WebSphere feature packs installation . . . 226

Appendix B. Workload partitions benefits . . . 227
Improvement of service level agreements . . . 228
Optimization of resource usage . . . 228
Highly granular control of resource allocation . . . 229
Control of security and privilege commands . . . 229
Simplified handling of software stack . . . 230

Abbreviations and acronyms . . . 233

Related publications 237
IBM Redbooks 237
Other publications 238
Online resources 238
How to get Redbooks 240
Help from IBM 240

Index 241

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX 5L™
AIX®
Argus®
BladeCenter®
Candle®
Chipkill™
CICS®
DataPower®
DB2 Universal Database™
DB2®
developerWorks®
Electronic Service Agent™
Enterprise Workload Manager™
eServer™
HACMP™
IBM®
IMS™
Informix®
Lotus Notes®
Lotus®
Micro-Partitioning™
Notes®
OMEGAMON®
OpenPower®
POWER™
POWER Hypervisor™
POWER4™
POWER5™
POWER5+™
POWER6™
PowerVM™
pSeries®
Rational®
Redbooks®
Redbooks (logo) ®
RequisitePro®
RS/6000®
System i™
System p™
System p5™
System z™
Tivoli®
Virtualization Engine™
WebSphere®
Workload Partitions Manager™
z/OS®

The following terms are trademarks of other companies:

InfiniBand, and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.

AppArmor, Novell, SUSE, the Novell logo, and the N logo are registered trademarks of Novell, Inc. in the United States and other countries.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

mySAP, SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

LifeKeeper, SteelEye, and the SteelEye logo are registered trademarks of Steeleye Technology, Inc. Other brand and product names used herein are for identification purposes only and may be trademarks of their respective companies.

EJB, Enterprise JavaBeans, J2EE, Java, JavaBeans, JDBC, MySQL, Power Management, Solaris, Streamline, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, SQL Server, Windows Server, Windows Vista, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Itanium, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

This IBM® Redbooks® publication gives a broad understanding of the synergy between service-oriented architecture (SOA)-based applications and Power Systems servers. The popularity and reach of SOA-based applications has grown exponentially in recent years. Enterprises are relying more on SOA-based applications for their operation. As a mission critical system, it is critical that the application be supported by an adequately planned infrastructure.

IBM Power Systems have been leading players in the server industry for decades. Power Systems provide great performance while delivering reliability and flexibility to the infrastructure. Given the advent of SOA-based applications, this book aims to demonstrate what benefits a SOA-based application can get from a Power Systems infrastructure and how Power Systems support a SOA-based application.

The book is intended as a guide for a Power Systems specialist to understand the SOA environment and for a SOA specialist to understand the facilities available for Power Systems supporting SOA-based applications.

Note: Although the term Power Systems encompasses both the IBM System p™ and IBM System i™ hardware, we use the term System p in this book to specifically discuss the System p hardware.

The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Figure 1 Left to right: Budi Darmawan, Christian Richter, Gustavo Lima, Mario Ernesto Orozco Salazar, Monty Poppe, and Vidyadharan Elambila Veetil

Budi Darmawan is a project leader at the International Technical Support Organization, Austin Center. He writes extensively and teaches IBM classes worldwide on all areas of systems management. Before joining the ITSO nine years ago, Budi worked in IBM Indonesia IT services department as lead implementor and solution architect. His current interests are SOA application management, availability management, and Java™ programming.

Christian Richter is an Associate IT Architect who joined IBM in 2006 after completing a one-year professional career training program at the Research and Technology Center North America of the Robert Bosch Corporation in Palo Alto. Christian majored in software engineering in 2005 at the Cologne University of Applied Sciences in Germany with a focus on operating systems and distributed systems. Currently he is part of the Dublin Software Lab team and works with IBM DB2® Universal Database™, WebSphere® Portal Server, and WebSphere Process Server. His areas of interest include Model driven software development and architecture, SOA, and distributed systems.

Gustavo Lima has more than 11 years of experience in large systems, storage networking, and critical systems support. He spent the first nine years working as an IT Solution Architect at some of the largest I/T, telecommunication, and banking sites in Latin America, Africa, the USA, and Asia, and was responsible for providing technical solutions to business problems. He joined IBM in 2005 as a Weblogic SME to lead the team responsible for providing Middleware Support. His areas of expertise include TCP/IP communications, Security, UNIX® and Linux® System Services, Weblogic, and WebSphere.

Mario Ernesto Orozco Salazar is a Senior IT Specialist in Mexico concentrating on helping customers design service-oriented architectures and implement Enterprise Service Buses. He works with Empresa Grafica Consultora S.C. in Mexico, an IBM Business Partner. He has eight years of experience in the IT field. He holds a degree in Electronics Engineering from ESIME Zacatenco, IPN. He has used his skills at customer sites working as a consultant providing expertise in solution architecture and design application development, migration services, performance evaluation, problem determination, project specification, and sizing.

Monty Poppe is a Software Engineer in Server & Storage System Test, where he configures and runs customer-like enterprise architectures using IBM System p servers. He graduated from Texas A&M University and has been with IBM since 2001. Before moving into Quality Assurance, Monty spent time as a developer and a consultant.

Vidyadharan Elambila Veetil is a SOA consultant from IBM India.

Thanks to the following people for their contributions to this project:

Scott Vetter, Octavian Lascu, Wade Wallace
International Technical Support Organization, Austin Center

Meghna Paruthi, John Langois, Sandy Amin, Julie Craft
IBM Systems and Technology Group

Francois Briant, Francoise Alabiso, Yann Kindelberger
IBM European Customer Center, Montpelier PSSC

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Power Systems and SOA synergy

This book discusses the synergy, implementation, and operational issues of running a SOA-based application on a Power Systems platform. This chapter provides an outline of the overall discussion of Power Systems and SOA. The discussion in this chapter includes:

- 1.1, “SOA basics” on page 2
- 1.2, “Power Systems and SOA” on page 3

1.1 SOA basics

The service-oriented architecture (SOA) environment provides a flexible application infrastructure. The SOA principles of loosely coupled applications composed of standards based components appeals greatly to companies that want a faster application development cycle and agile business processes.

SOA is an architectural style that supports service orientation. The business is viewed as a series of linked services where applications are loosely coupled through standard interfaces.

SOA mirrors existing business logic as a workflow of services This allows business processes to be constructed and re-constructed using an assembly of services. SOA concepts and architecture will be covered in Chapter 2, “SOA introduction” on page 5. We explore the SOA life cycle, SOA components, and IBM SOA Foundation model in that chapter.

This book aims to introduce the benefits of using IBM Power Systems as the server platform of choice for SOA-based environment implementation. There are different ways to construct a SOA solution. However, there are some common inherent features of the SOA:

- Distributed application
- Loose coupling of service components
- Service component reuse and rearrange
- Unpredictable workload demands on the IT Infrastructure

Because services can be linked and rearranged quickly to meet new customer requirements, the demands on an IT Infrastructure are great. Without a flexible, scalable, reliable, and secure server infrastructure, the SOA model collapses. Infrastructure selection for a SOA environment is discussed in Chapter 3, “SOA infrastructure selection” on page 73.

You know that your server IT Infrastructure does not properly support the SOA model when:

- Your servers grow out of their available space.
- Your dedicated servers average low utilization rates.
- Your electric bill becomes a monthly focus item.
- The cost of managing the server farm is greater than the cost of developing new software.
- Software licensing costs increase rapidly.
- Service level agreements are broken at an unacceptable rate.

- You experience security breaches through the server or operating system.

Power Systems are designed with tightly integrated components (server hardware, an operating system, and PowerVM™ Virtualization) to accommodate the unpredictable demands of a SOA environment. When you choose your server IT infrastructure, consider these important SOA demands:

- Flexibility: Services can be called by existing programs or any new programs. The actual processing loads are not determined in the development cycle. Since you cannot predict the workload, the server hosting the services must be flexible enough to accommodate additional load by adding through capacity on demand, virtualization, load balancing, live partition and workload migration, and other means.
- Scalability: The dynamic nature of the workload means that systems demand and configuration are changing, which requires servers to adapt to the changing environment. Energy usage can increase when additional servers are acquired. Conserving server capacity by consolidating demand to a small number of servers helps with management and the ability to adapt.
- Reliability: Because applications are loosely coupled, they will not dynamically check service availability. Services are expected to exist and remain available whenever called. The server infrastructure must be reliable in order to meet service level agreements. Power Systems uses mainframe inspired reliability, availability, and serviceability features to support the SOA environment.
- Performance: The nature of a SOA environment is large fluctuations in the workload. Services should perform in an acceptable amount of response time, otherwise the caller may assume that the service is not available and perform fault processing.
- Security: Data must be protected as it flows through services at all points in the network. The server, operating system, and network must be designed to provide information assurance.
- Governance and manageability: The distributed nature of services requires additional complexity to keep track of all the services. It also amplifies the need for good monitoring, as problem determination may be complicated because it encompasses different systems, networks, and platforms.

1.2 Power Systems and SOA

IBM System p has been the IBM flagship for an UNIX-like environment. The original RS/6000® family of machines became IBM eServer™ pSeries® and IBM System p servers, and with each evolution, we have seen growing server reliability and performance.

Recently, IBM created the term *Power Systems* to refer to the combined family of System p and System i servers. Both families can run on the same hardware platform, signifying the convergence of technology. This IBM Redbooks publication is aimed at addressing Power Systems running AIX® and Linux operating systems. Therefore, we may use the term *System p* to refer to this fact.

Power Systems allows mainframe level reliability with redundant components and also provides hardware virtualization to provide cost effective flexibility. Power Systems features are discussed in Chapter 4, “IBM System p features” on page 99.

IBM Power Systems strengths include:

- The fault tolerant reliability, self healing, and self recovery features of the IBM Power Systems design allows outages from hardware problems to be minimized.
- Processing performance that provides high price-performance ratio, which can be seen from benchmarking results found at the following Web site:

  `http://www.tpc.org/tpcc/results/tpcc_perf_results.asp`
- Virtualization that allows consolidation of servers into a single hardware. It facilitates processor pool sharing and hardware redundancy. A virtualized environment provides a flexible and cost effective solution for implementation.
- Hardware and software management using Hardware Management Console, enterprise Workload Manager, IBM Directory, and various Tivoli® monitoring and provisioning tools.

IBM System p runs AIX or Linux on POWER™. Most of the hardware benefits are reflected in the operating system implementation. SOA middleware software from IBM is available on both the AIX and Linux operating systems. We explain implementation considerations, from hardware setup to middleware installation and configuration, in Chapter 5, “SOA implementation” on page 167.

Additionally, we also discuss a sample operational scenario for SOA application in Chapter 6, “Operational environment” on page 197. This chapter explores a sample application and how the application is handled in a System p environment.

We also provides additional information in Appendix A, “WebSphere Application Server and WPAR” on page 221.

2

SOA introduction

In this chapter, we discuss general information about the service-oriented architecture (SOA) in order to avoid a misinterpretation of the term. We also explain aspects of SOA that are necessary to understand the rest of the book. The key elements we cover in this chapter are:

- 2.1, “Understanding SOA” on page 6
- 2.2, “Benefits of SOA” on page 16
- 2.3, “Common misconceptions” on page 17
- 2.4, “SOA standards, mediation, and registry” on page 19
- 2.5, “IBM SOA Foundation” on page 39

2.1 Understanding SOA

There are varying SOA definitions across different roles within an enterprise. For example:

- For a business executive and business analyst, SOA is a set of business services provided by IT assets that can be used to build solutions for customers and partners.
- For an enterprise architect, SOA is a set of architectural principles and patterns addressing overall characteristics of solutions: modularity, encapsulation, loose coupling, separation of concerns, reuse, composability, and so on.
- Similarly, project managers, testers or quality assurance engineers, and software developers have their own interpretations of the term.

In fact, software products that position themselves to be service-oriented or support SOA possess different views of SOA too. This contributes to the confusion surrounding SOA. None of these definitions are wrong, though, as SOA can have a different impact for each role, leading to different definitions about what SOA can mean for this role. However, these different definitions may lead to some misunderstandings. We discuss some of them in 2.3, “Common misconceptions” on page 17.

In this chapter, we start introducing the origins of SOA, look into what a service actually means from a business and IT perspective, and show the basics of service orientation thereafter.

Many resources for SOA related topics and implementation examples can be found at the IBM developerWorks® SOA Web site at:

http://www.ibm.com/developerworks/soa

Additionally, the Object Management Group (OMG) has a Web site that provides interested individuals a place for discussions, terminologies, methodologies, and models for further reference across SOA topics. The Web site can be found at:

http://soa.omg.org/

2.1.1 The evolution towards SOA

Figure 2-1 on page 7 shows the evolution of programming paradigms towards SOA.

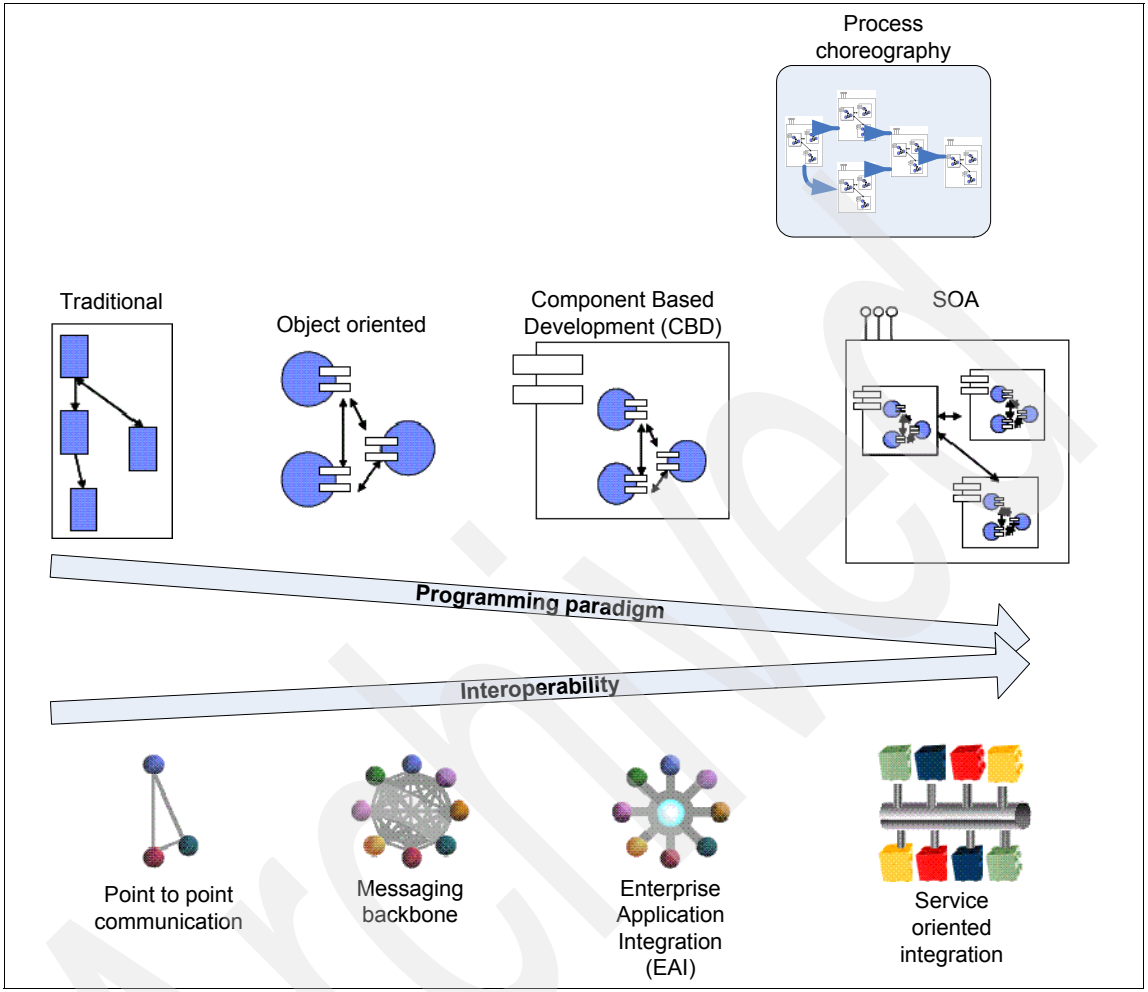


Figure 2-1 Evolution towards SOA

In Figure 2-1, the programming paradigm evolves from traditional procedural programming to object orientation, then component based development, and finally to service orientation.

This evolution also enhances the way programs communicate with each other. Traditional programs often have proprietary mechanisms to communicate with each other, hence providing point to point communication that is hard to extend. Object orientation and component based development comes with the potential to have a messaging backbone that allows a more flexible communication structure. It also develops the Enterprise Application Integration infrastructure.

The service orientation allows Enterprise Service Bus to serve as a common bus for communication between components.

As shown in Figure 2-1 on page 7, object orientation encapsulates a programming method and data. This encapsulation is enhanced by component based development to achieve a better structure for reusing the components. Service orientation encapsulates the components that provide a standard-based interface for interacting with other components. This shows that SOA is not a new style of programming: it is an architectural method composed of a solution and its components.

SOA delivers on the promise of a more flexible business through creating more flexible IT around business solutions. The fundamental premise of SOA is that if the business design can be described and maintained, there is an immense potential for leveraging this information in the design of IT around this business design to make IT an effective supporting part of the business design.

There are standards for interoperability, such as Common Object Request Broker Architecture (CORBA), Remote Procedure Calls (RPCs), and Remote Method Invocation (RMI). However, the complexity and proprietary nature of the technology and their specification leads to Web services standards.

The wide acceptance of Web services standards and their associated technologies can be seen as one of the driving forces towards the SOA paradigm. Many technology standards for Web services are implemented using the XML structure. This also accelerates its acceptance. These provide a powerful set of features based on open standards and technology for delivering on the promise of SOA

In 2.4.1, “SOA standards” on page 20, we discuss more standards that elevate SOA to the position of a powerful architectural concept that can narrow the gap between business and IT.

2.1.2 Defining services

Services are the key building blocks of SOA. To define SOA, we must define the concept of a service. A service is a reusable function that can be invoked by another component through a well-defined interface.

The main role of a service is to exploit important business services in a flexible, easily composed, and highly reusable fashion. This means that not all application functions should be exploited as a service.

In general, we can define a service as a task within a business process. A business process can be broken down into a series of services. The granularity level of the service can vary:

- An elementary decision can be defined as a service, such as whether an invoice is paid or not.
- Business rules can be implemented by a service. An invoice can be considered as paid if a hold using a credit card is in progress.

It is important that services are identified, designed, and realized in the right way. It assists the vision of business agility through IT flexibility. In this section, we talk about the major service design principles that promote IT flexibility by increasing both decoupling and ease of process implementation. Figure 2-2 shows the service principles.

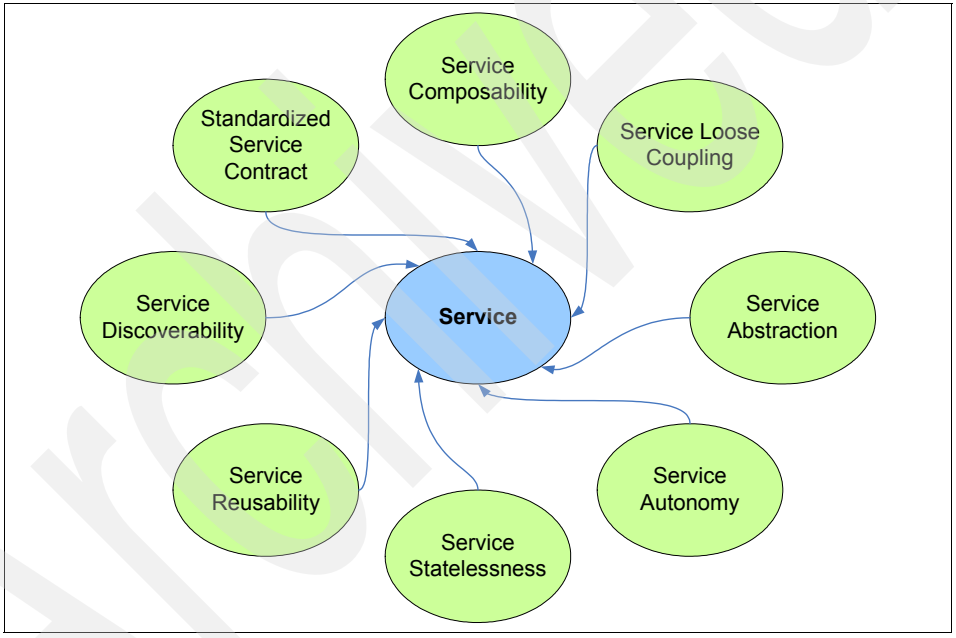


Figure 2-2 Service design principles

Standardized service contract

A service contract can be specified between the service provider and service consumer. Such a service contract is a formal description of the service from a technical and non-technical perspective. A service contract captures a service's behavior, interface, policies, and service level agreement (SLA). It contains everything that a requestor and provider of a service must agree upon so that the requestor knows what it can expect and a provider knows what it must do.

Contracts may refer to business requirements and technical standards such as “A shipping provider may provide its premium customer with a shipping service contract of shipping goods to that customer in less than 12 hours across the entire country”.

Service contracts are the basis for service design. They also allow more decoupling of the service provider and consumer by identifying the service behavior clearly. A service contract can consist of:

- Functional requirements, that is, what the service does and what result it returns.
- Non-functional requirements, such as invocation constraints or conditions.
- Service Level Agreement and security policies

Service loose coupling

Services are loosely coupled, that is, they hide their implementation details and only expose their interfaces. In this manner, service requesters do not need to be aware of any underlying technology or programming language the service is using.

The looser coupling between services allows for a quicker response to changes than is usual with applications. This results in a much faster adoption to new business needs. SOA also leads to a much more effective architecture and design of components, as they are designed specifically around an issue.

To enable the interaction between services, the services require a defined and standardized interface. The standard must allow a broad range of scenarios to be covered and must be flexible to handle additional requirements in the future.

Using loose coupling and the service registry, it is easier to add, remove, and move services to adopt new versions or configurations. The business process would not be interrupted by service changes as long as the changes is reflected in the service registry. Figure 2-3 on page 11 illustrates a service relocation from two example domains, Domain B to Domain C. This concept is similar to the use of the Domain Name System (DNS) to map host names to IP addresses: Clients just refer to the host name, not the physical IP addresses.

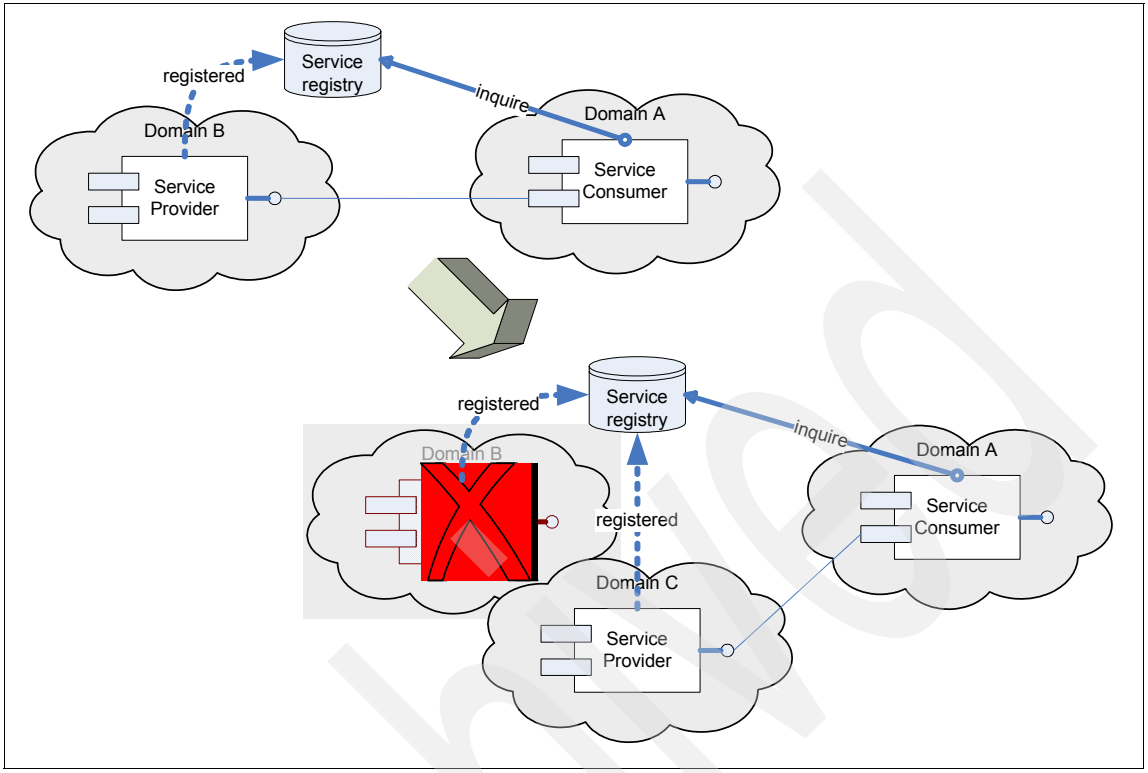


Figure 2-3 An example of a service relocation

Service abstraction

A service is abstracted in a contract where its functionality is defined. Even though this is considered a high level of abstraction, the details of this contract are limited to the underlying implementation. The service abstraction hides any implementation logic and only exposes the necessary interface for the service to the outside world. Abstraction also serves as a base for loose coupling.

Service reusability

A service encapsulates logic with a defined level of granularity. The level of granularity determines whether the service can be reused. A large-grained service is useful, but it is mostly not reusable. A very fine-grained service can be reused, but it may take several service calls to perform a function. Implementation must consider the correct level of decomposition to achieve reusability. The logic should be divided up to a level where reuse across the enterprise is possible and beneficial.

Service composability

The service composability principle means that collections of services can be coordinated and assembled to form new composite services. Composite services are easily built when services have consistent data types and formats. It also depends on the abstraction of the logic it encapsulates.

Service autonomy

Service autonomy is when services have complete control over the logic they encapsulate. A highly autonomic service can operate on its own without needing the surrounding services to perform the main functionality. On the other hand, a non-autonomic service cannot work independently from its surroundings and may therefore be limited in reuse. For services that are composed of other services, autonomy is achieved from the autonomy of the services it encapsulates and the logic of its own main functionality.

Service discoverability

Service discoverability demands that services can be defined and discovered for invocation. This principle is related to the service contract and service registry concept. The service registry allows services to be found and to get their service contracts. Discovery of services is also beneficial for reuse; a service registry may indicate that a service with the function already exists and can be reused.

Service statelessness

Service state information must be minimized to avoid excess usage and locking of resources during run time. This can prevent performance bottlenecks. Adhering to this principle will also increase the reuse potential and scalability aspects of the service.

2.1.3 Defining SOA

SOA is defined by the IBM SOA Center of Excellence as follows:

Service Oriented Architecture is an enterprise-scale IT architecture for linking resources on demand. These resources are represented as business-aligned services which can participate and be composed in a value-net, enterprise, or line of business to fulfill business needs. The primary structuring element for SOA applications is a service as opposed to subsystems, systems, or components.

The ability to link resources on demand is often hampered by incompatibilities in the enterprise application. Applications utilize proprietary formats for data, proprietary, or hybrid communication channels, and non-sharable configurations. They also may just address domain or department specific issues rather than

addressing an enterprise wide support of the business design. Organizing these applications to compose a set of distributed capabilities for the business is no easy task.

Without SOA, most systems in an enterprise are built on a “as you go” basis. When there is a need for an application, for example, an Order management system, the application is either procured or built. This happen for all applications within the enterprise. Hence, a situation rises where data (even about a single entity) resides across various applications within the enterprise. Interfaces must be built so that the marketing department can leverage the customer purchase history that resides in the order management system. A similar situation happens for other applications as well. The complexity increases exponentially when we consider the proprietary formats for data, interaction, and configurations across these applications. Also, traditional applications are usually designed in different layers that are not loosely coupled. Defining services entities are a challenge in itself.

Considering the different perspectives involved in SOA, it is best to consider SOA as a team sport that involves many different players from many different disciplines and business domains. All are working together to achieve a common goal: the alignment of business and IT.

We discuss IBM SOA Foundation in 2.5, “IBM SOA Foundation” on page 39. It explains the integration patterns for modeling an enterprise to SOA.

2.1.4 SOA basics

As discussed before, SOA is composed of services that encapsulate business tasks that work together to achieve a common business goal. This means that a SOA based architecture yields to composite applications, as applications are built from sets of integrated or related services.

Figure 2-4 shows the basic SOA components for creating SOA solutions. It does not consider implementation specific requirements like security or quality of service.

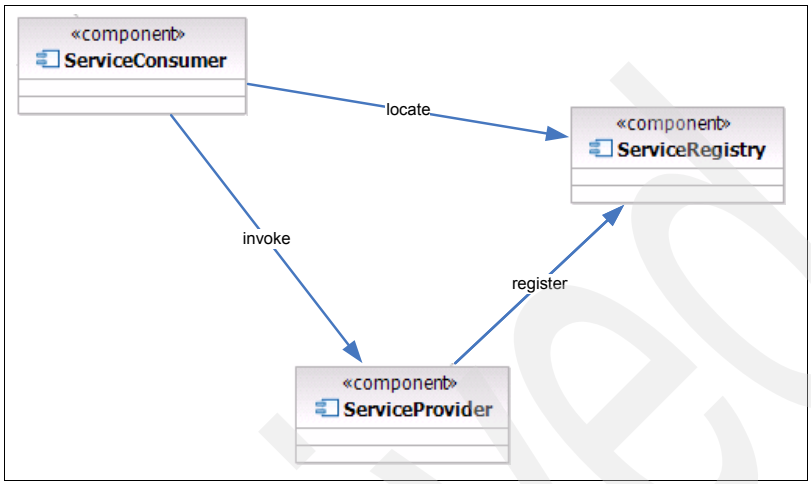


Figure 2-4 The basic components of SOA

Based on Figure 2-4, these are the components of SOA:

- Service registry

  The service registry allows other components to find services required for their operation. It supports static or dynamic registrations and publishes services known to the registry and stores their meta data.

- Service provider

  Each service provider provides a defined set of functionality regarding a business need. For example, for the business need to send an invoice after an order has been received and the shipment is prepared, a service provider can be defined to send invoices based on the order data it receives.

- Service consumer

  A service consumer is the component that invokes a service provider and uses its service. Note that a service provider can also invoke other services, thus becoming a service consumer too. For example, the send invoice service can invoke other services to collect information such as customer contact information and order status.

We highly recommend using a standard service registry even in small environments to ease adoption and discovery across an enterprise. Some development tools, like IBM Rational® Software Architect, allow the integration of registries in the modeling and development process to prevent a

re-implementation of an already existing services. This approach ensures services are known at an early stage, which increases reusability.

Besides the known relationships, a specific service contract may define the relationship between the service provider and service consumer. This gets more and more important as the SOA spreads across different business units, departments, and countries. Services get out of departmental influence and control. Service Level Agreements may be defined to ensure that the service requirements are known.

Service granularity defines the level of abstraction chosen for the service. A fine abstraction granularity for a service may be, for example, a service that just looks up a reference number for a book, while a higher abstraction may refer to a service that performs whole business operations, with the possible use of many other services on lower abstraction levels.

There are various degrees of adoption of SOA, such as:

- Implementation of individual services, where the usage of SOA concepts is limited when building new services or wrapping existing services to make them available as SOA components.
- Service integration, where a business composes services through business process flows or state machines to implement complex business processes.
- Enterprise wide IT transformation, where SOA concepts are adopted enterprise wide and pervade application deployments enterprise wide.
- On Demand Business Transformation, where SOA transforms business processes from the real world towards the IT domain and vice versa.

Summary: SOA is based on a simple but powerful service provider and service consumer model for designing and developing components and building applications or business processes.

2.2 Benefits of SOA

Based on the definition of a SOA application in 2.1, “Understanding SOA” on page 6, the following items are benefits of SOA:

- Business agility and flexibility

  A major benefit of SOA is the positive impact on business agility and flexibility. It accelerates the time of response to business needs and realigns the whole IT infrastructure. Being able to compose business solutions out of services allows for a great opportunity to build innovative solutions and deliver new business functions rapidly. This also leads to more time spent with the delivery of new functions rather than with environments and partially redundant components.

- Cost saving and reusability

  Reusability is a major part of SOA and is implied by the use of components. Each service consumer may act like a service provider on the other side and provide its functionality for further reuse. An example is a service for ordering new parts for a production environment. This service may be directly used by a person using a front-end application, but the same service may be part of the integrated supply chain management system.

- Model driven system development

  Another benefit for IT is the creation of new services and assets. Through modeling techniques and model driven systems development, the work of introducing new services can be significantly reduced. See *Model Driven Systems Development with Rational Products*, SG24-7368 for more information.

- Other benefits that can be easily reached using SOA are:
  - Deliver role based interaction and collaboration.

    It is easier to perform collaboration and deliver content based on the role of a person. A centralized role and authentication mechanism can act as a service for the enterprise wide directory server.

  - Achieve business process innovation by treating tasks as modular services.

    Business processes become more dynamic as the underlying tasks are encapsulated within services. A task is defined by its functionality; the implementation is done under a service interface. Changes in a specific tasks require no changes in business processes and vice versa.

– Provide trusted information in a business context by treating it as a service.

  By encapsulating information as a service call, context specific information can be provided for different invocations instead of providing the same static content.

– Service enables existing assets and fills portfolio gaps with new reusable assets.

  Existing solution or parts of existing solution can be masked to enable a service interface. These masks (facades) allow existing solutions to participate in a SOA solution.

  Note: Creating a facade of existing assets may require a conversion of data and meta data to fully integrate them in a flexible and independent service-oriented infrastructure.

– Connect and integrate systems, users, and business channels based on open standards.

  Integrating with external business systems will get much easier by just accessing services. This allows integration with external Business Partners and solution providers into the current IT infrastructure and existing business processes.

The above list of benefits are not exhaustive, but the list should give you an idea of the ability of a SOA solution.

2.3 Common misconceptions

We introduce some common misconceptions as seen from a couple of different roles. These can manifest as barriers to an efficient SOA implementation and adoption throughout an enterprise. The focus of this section would be to highlight a few common misunderstandings in each of the adoption, design, and realization phases.

- SOA is a technology thing!

  This is a common perception among some business analysts and executives. SOA is perceived as something for technical folks. Hence, their corporations embark on a SOA effort from an IT perspective instead of a business idea. Implementations might be technically feasible and sometimes successful, but the impact on the business may not be realized since it was never considered in the first place.The cost of IT rises without realizing any return on investment.

- Web services is SOA.

  Web services are an established and well standardized technology that helps build service-oriented architecture, but they may not necessarily lead to one. Web services themselves can be seen as one possible type of connection between services. But seeing Web services as a path to SOA will lead to issues. Building or integrating a SOA in an enterprise needs more than just defined connection points.

- Let us convert our back ends and data structures to XML to get a SOA.

  Converting the data does not lead to a new architecture. It may ease interaction between components, but without an appropriate design and structured approach, this effort leads to nothing else than work. XML is highly used in a SOA environment, but you must be certain that services make appropriate use of it. Taking XML from back ends and converting them to a native format adds extra processing, which may lead to a performance bottleneck. It may also be more difficult to use.

- Let us buy this new SOA enabled product and we will have SOA in our enterprise

  We often hear, “We have a WebSphere Process Server implementation. We are a SOA enterprise”. Indeed, this will bring possibly (one) SOA element into an enterprise, but that it is all. SOA is not a product! Unless the product is leveraged by a SOA architecture, it will not lead to any business benefits that SOA promises to deliver. An application server, for example, may offer SOA oriented connection points a developer can utilize to manage and configure applications, but without the appropriate infrastructure or architecture taking benefit of this functionality, it will not provide any business benefits. On the other hand, a product or product selection might be considered to be the first step in a SOA integration, but this should not be the only consideration to take into account. You should always analyze if and how a product will add a step towards SOA.

- Let us ensure all our business processes are captured and run in business process execution language (BPEL) to prepare for SOA.

  This action might support the establishment of SOA to a small extent, but it will result in work that may not be required or may not be achievable in a timely way. Aligning to a standard will not support SOA, as a standard is only designed for a specific and usually limited purpose. BPEL, for example, is an XML based language that specifies business processes’ execution and business interactions. Following a more structured approach for getting started with SOA will prevent unnecessary work and enable reuse across business processes where some of the existing processes might be already covered by other processes.

- I do not need to apply governance to a SOA implementation.

  Just like an enterprise cannot run without proper management, SOA also cannot. In fact, SOA is closely linked to the business. It is a new way of looking at business and how it operates. So, without governance, a SOA realization will just fail. A typical problem an enterprise would run into when utilizing SOA without governance is a proliferation of services: Because there is no governance framework for knowing where the services are or how to use them, the reusability and composability of these services are just not possible. Hence, new ones delivering the same functionality are built, leading to duplication. The benefits of SOA are lost.

There are many other misunderstandings that can be seen across all levels, management, architecture, implementation, and infrastructure. Most of them result from the assumption that the usage of open and new standards will necessarily lead to the new architectural style of SOA, which it will not. All those misunderstandings do not necessarily come from a misunderstanding of SOA principles, rather, it stems more from architectural thinking about a solution and its components. A wrong approach may prevent leveraging the full benefits of SOA and can introduce common mistakes.

The developerWorks article *SOA Antipatterns* can serve a quick guide to see what not to do or how to correct such mistakes. Refer to the following Web site:

`http://www.ibm.com/developerworks/webservices/library/ws-antipatterns/`

To help design and leverage SOA to its full extent, 2.5, “IBM SOA Foundation” on page 39 is a good overview of a successful integration or new solution based on service orientation. The section introduces the SOA foundation with its life cycle and reference architecture. It also helps guide the thinking process towards SOA in the form of an agile and versatile IT infrastructure with the purpose supporting the business.

2.4 SOA standards, mediation, and registry

This section focuses on explaining the guiding principles and standards on which SOA is based. The discussion includes:

- 2.4.1, “SOA standards” on page 20
- 2.4.2, “Enterprise Service Bus and SOA” on page 22

2.4.1 SOA standards

When building a SOA-based application, the services must operate together to perform the overall business function. When the application connects many different service providers and consumers, standardization becomes an obvious requirement. The larger the environment is, with more services to integrate, the more important adherence to standards are.

SOA should be seen as an architectural style. This does not imply any technical standards for its implementation, such as a programming language, server platform, or communication mechanism. A SOA based solution should not be based on proprietary or vendor specific APIs. Using industry wide and open standards enhances the flexibility of the solution based on SOA principles.

Currently, and for some time to come, many of the technologies that are used to implement SOAs are evolving. Therefore, individual SOA solutions must make carefully balanced decisions among customized, proprietary, and open-standard technologies, which characteristics and components of SOA to implement, and which areas of business functions and processes to apply them to. Of course, the decisions between business benefits, technology maturity, and implementation or maintenance efforts should be balanced.

Figure 2-5 on page 21 shows the various standards and specifications required for the enterprise adoption of SOA.

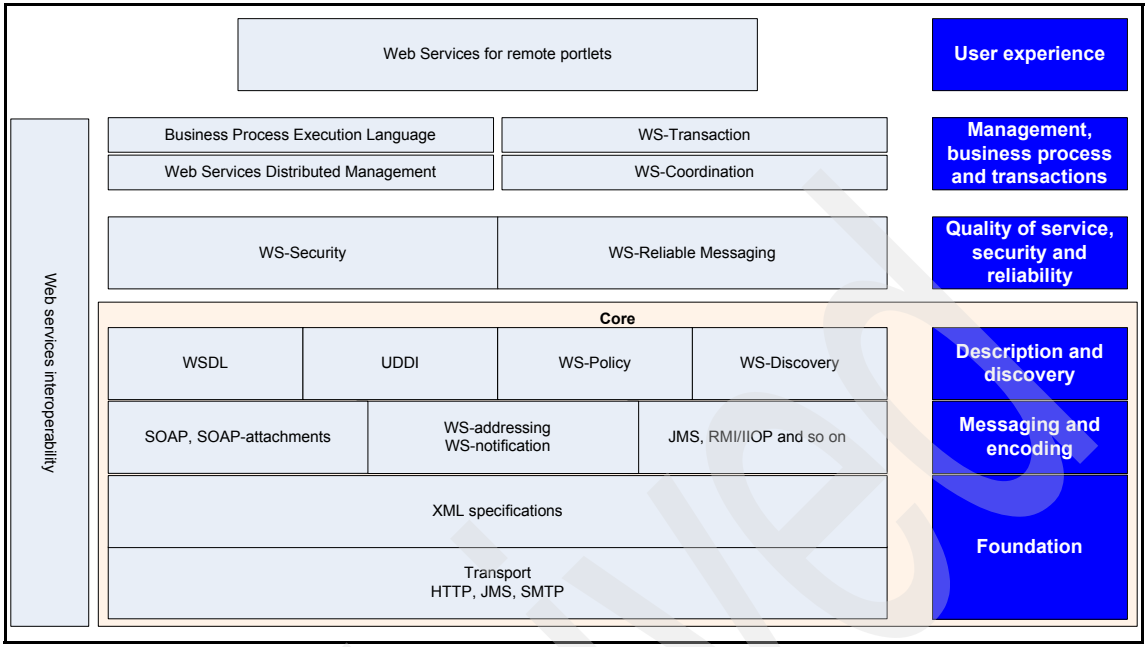


Figure 2-5 Standards and specifications

Figure 2-5 is divided into discrete domains. The domains are shown in the right most column. These standards allow disparate systems to communicate with each other without the use of proprietary technologies. As mentioned earlier, some of these standards are still evolving and a measured approach to adoption is advised. Some of the standards are considered core elements, as they are part of the basic implementation of a SOA environment:

- Foundation standards define the transport mechanism, including HTTP, Java Messaging Services (JMS), and even SMTP, and the use of the XML specification as the basic encoding mechanism.
- Messaging and encoding standards enforce how messages are transported. Most of the implementation uses the Simple Object Access Protocol (SOAP) standard, which also includes addressing and notification mechanisms.
- Description and discovery standards determines the policy and discovery mechanisms. This also defines the metadata structure in the Web Services Description Language (WSDL) specification.
- The quality of service, security, and reliability standards defines the security (WS-security) and reliability (WS-reliable messaging) standards.
- Management standards include additional management, governance, coordination, business processes, and transaction specifications.

- User experience standards defines the portlet interface.
- Interoperability across different domains is defined by the Web Services Interoperability organization. More information about this topic can be found at the following Web site:

  http://www.ws-i.org

The framework defined in the specification enables an application service to create the context needed to propagate an activity to other services and to register for coordination protocols. It enables existing transaction processing, workflow, and other systems for coordination to hide their proprietary protocols and to operate in a heterogeneous environment.

There are many different specifications to cover specific issues in regards to distributed applications based on Web services. Visit the IBM developerWorks home page about Web services standards for further reference and information at:

http://www-128.ibm.com/developerworks/webservices/standards/

2.4.2 Enterprise Service Bus and SOA

An implementation of SOA requires applications and an infrastructure that supports the SOA principles. Applications can be enabled for SOA by creating service interfaces to existing or new functions, either directly or through the use of adapters.

Another approach can be the usage of the facade pattern in conjunction with a service provider enablement (see *Design Patterns: Elements of Reusable Object-Oriented Software*, by Gamma, et al, for more information about the facade pattern).

An infrastructure for SOA, at the most basic level, involves provisioning capabilities to connect, route, and deliver messages between service consumers and service providers. The infrastructure should support the SOA principles, in particular, the ability to provision for loose coupling. It should have the ability to substitute the service provider with no effect on the service consumer, that is, the ability of the service consumer to invoke services in a manner independent of the service location and the communication protocol involved. As organizations expose more and more functions as services, it is vitally important that this infrastructure also supports the management of SOA on an enterprise scale.

The Enterprise Service Bus (ESB) offers such services. It can provide routing, substitution, and unified management of SOA communication. ESB developed from the need for a clear integration through a defined and standardized middleware layer for SOA. ESB products do not create SOA out of the box, but

they do support its implementation through a variety of standards they support. It combines the major enterprise integration patterns in use today into a single entity. Figure 2-6 shows the updated SOA components structure with ESB.

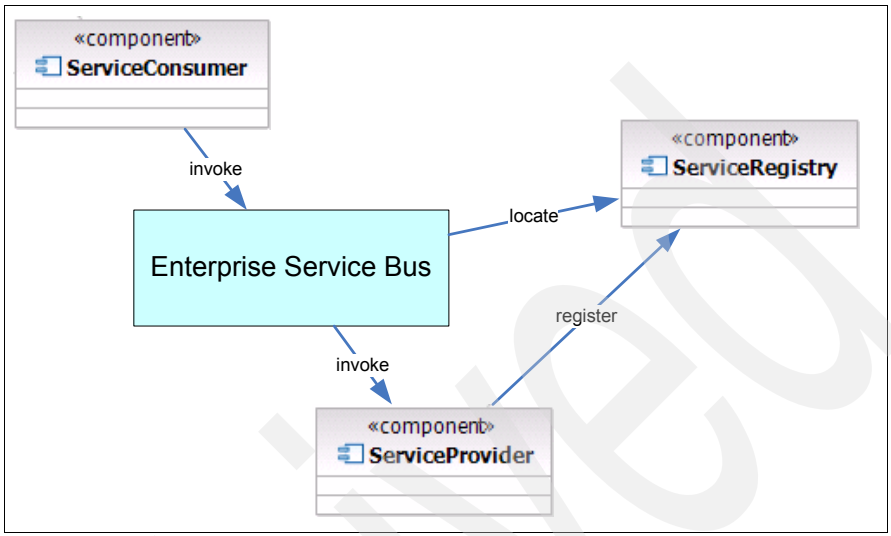


Figure 2-6 SOA components with ESB

ESB provides an infrastructure with the following key abilities:

- Decoupling of the consumer from its implementation

  Decoupling the consumer's view of a service from the actual implementation greatly increases the flexibility of the architecture. It allows the substitution of one service provider for another (for example, because another provider offers the same services for lower cost or with higher standards) without the consumer being aware of the change or without the need to alter the architecture to support the substitution.

- Decoupling of service interaction aspects

  The services are decoupled from communicating directly with each other by talking through an intermediary. The intermediary maps the location of the request towards the final service provider endpoint.

- Integration and management of services

  New services can be easily integrated and managed across the whole ESB. This allows for easy addition and referencing of services. For integration purposes, a common pattern can be used based on the ESB's standardized interfaces. See *Patterns: SOA with an Enterprise Service Bus in WebSphere Application Server V6*, SG24-6494 for more information about this topic.

An ESB provides a layer that removes any direct connection between service consumers and service providers. Consumers connect to the bus, which acts as an intermediary, and not to the provider that actually implements the service. This type of connection further decouples the consumer from the provider. A bus also provides additional value add capabilities. For example, security and delivery assurance can be implemented centrally within the bus instead of having them buried within the applications. From a software engineering perspective, we can see the ESB as a proxy of the final endpoint with providing an abstraction that hides the location and implementation of the endpoint.

Integrating and managing services in the enterprise outside of the actual implementation of the services in this way helps increase the flexibility and manageability of SOA.

Figure 2-7 shows the benefits of an ESB topology compared to a point to point service interaction.

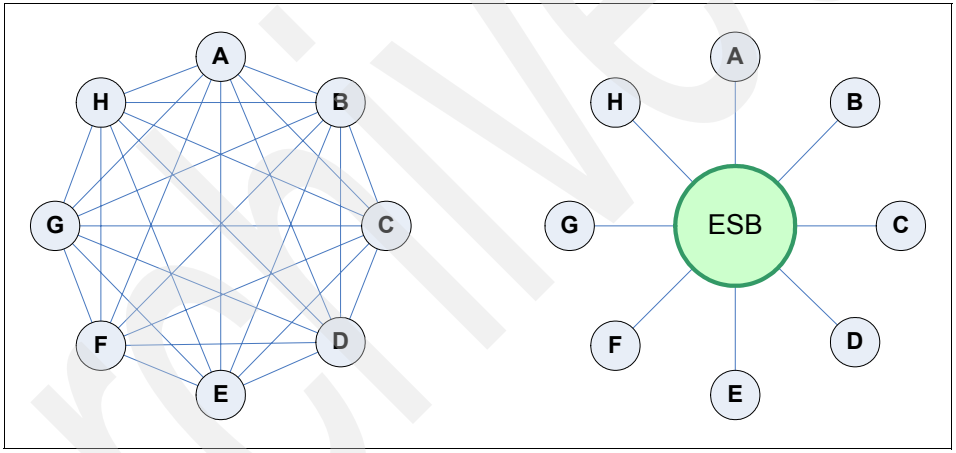


Figure 2-7 Point-to-point service interactions

The point-to-point interconnection in Figure 2-7 indicates that each service has to know its peers and manage its own communication method to its peers. The ESB integration makes sure that when a service is plugged into an ESB using an adapter, it is integrated with all other services on the ESB without having to be concerned with the particular communication protocol or interface exposed by these other services.

The Enterprise Service Bus concept is often used to describe the mechanism by which services interact with consumers and each other. The ESB provides the connectivity infrastructure and connects and integrates enterprise applications. The application can be in different locations, in deferment platforms across

different transports, and across organizational boundaries. The main concepts of an Enterprise Service Bus are:

- A consistent, location-transparent, and protocol-independent means to address services.
- A service endpoint that is only concerned with connecting to a local communication protocol, such as opening an HTTP socket or to place or retrieve a message from a local queue.
- ESB will communicate service request-responses through whatever mixture of protocols (HTTP, messaging, and so on) that provide connectivity between any set of endpoints.

Loose coupling

One of the main premises of SOA implementation is loose coupling. Though loose coupling might require more up front work, it creates flexibility and seeds the foundation for efficiency. Loose coupling must be designed carefully and sometimes it requires additional implementation work. The resulting flexible services can be reused in future applications.

Tighter coupling makes changes difficult and possibly expensive to implement. Tighter coupling tends to cost more over time because of the following factors:

- Synchronizing multiple organizations with the change.
- Adapting and redeploying updated components without affecting others.
- Making changes is hard and expensive, or in some cases impossible.
- Knowledge is distributed throughout the code.
- The same people are solving business and infrastructure problems.
- Different parts of solution are difficult to manage separately.
- It is hard to move, hard to scale, hard to distribute, and hard to replace.
- More coupling implies more expensive testing.

The degree of coupling for the service consumer and provider is directly proportional to the flexibility of the service. Looser coupling adds cost in terms of infrastructure and performance, but tighter coupling is expensive in terms of changes and maintenance. A balance must be made based on the overall needs:

- Service interface

  SOA emphasizes loose coupling by requiring that the service interface be described in a formal implementation-independent manner. The service interfaces should be published in a service registry. Publishing the service interface in a service registry promotes the reuse of a service and enables a service customer to perform early or late binding to such services.

- Business data models

  These models are usually coupled. There needs to be a mutual understanding of domain concepts, such as a Customer, Account, or Order.

- Communication protocols

  Services should use a protocol-independent service interface. This allows the protocol binding in the service interface definition to be changed. An example is to change SOAP/HTTP to SOAP/JMS, which should not change the service implementation.

- Data formats

  Services should be able to understand and work with any format. Middleware transformation capabilities can be used in the service infrastructure to perform the required transformations without affecting application code or behavior.

- Service provider anonymity

  The identity of a service provider should be negotiated through a third-party broker component.

- Platform independence

  A service consumer or provider should not be aware in any way of the hardware, operating system, or even application server platform supporting the others.

Loose coupling in a point-to-point interaction means that interface modules must be defined and coded for each new type of connectivity. New interface development can be very high in a large enterprise. Using the Enterprise Service Bus, these interfaces are integrated within the bus, as shown in Figure 2-8.

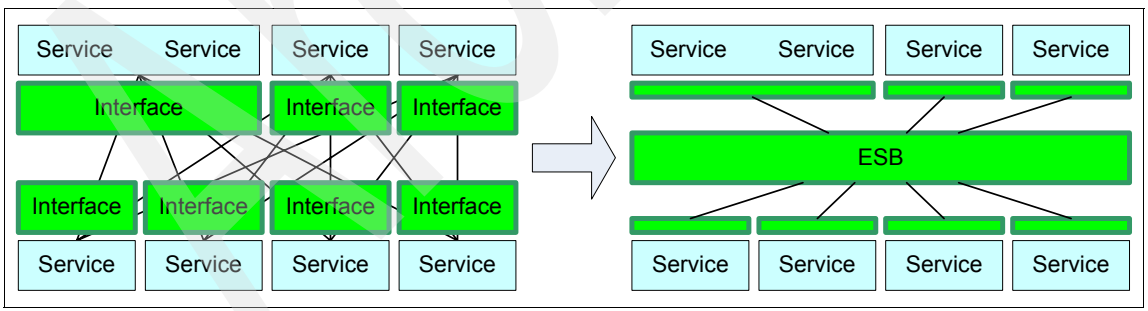


Figure 2-8 Loose coupling with ESB

The Enterprise Service Bus:

- Decouples the point to point connections from the interface
- Allows for dynamic selection, substitution, and matching.
- Enables more flexible coupling and decoupling of the applications.
- Enables you to find both the applications and the interfaces for reuse.

ESB brings flexibility by changing the service without impacting the service consumer and reusing existing services to build a new business service. ESB enables us to add new services faster by just defining it to the ESB. It also allows changing services with minimal impact to existing services, such as providing translation from the old service to the new service.

IBM ESB offering and positioning

IBM offers several ESB offerings.They are:

- WebSphere Enterprise Service Bus
- WebSphere Message Broker
- WebSphere DataPower® Integration Appliance

IBM WebSphere Enterprise Service Bus

IBM WebSphere Enterprise Service Bus (ESB) V6.1, supports the following key capabilities and benefits:

- A platform-based ESB

  WebSphere ESB supports advanced Web services, including SOAP/HTTP, SOAP/JMS, WSDL, and Web Services Gateway. It adheres to WS-* Standards, including WS-Security and WS-Atomic Transactions, and includes an UDDI Registry that can be used to publish and manage service endpoint metadata. It has pre-built mediations and supports hundreds of ISV solutions with extensive WebSphere Adapter support, including JCA-based adapters.

- Ease of use and business flexibility

  WebSphere ESB supports pattern-based configurations, improving productivity for administrators configuring WebSphere Application Server Network Deployment clusters and all required resources for WebSphere ESB deployment environments. It has an Eclipse-based profile management tool that is consistent for all WebSphere platforms. WebSphere Integration Developer provides an integrated, interactive, and visual development environment for rapid development of integration logic that requires minimal knowledge of Java or J2EE™.

- Integration with connectivity and transformation capabilities

  WebSphere ESB enables integration with any third-party JMS 1.1 Application Server Facilities (ASF) compliant messaging provider and automatic setup of WebSphere Application Server generic JMS resources. It supports generic HTTP interactions with modern RESTful service clients and existing Web services.

  WebSphere ESB delivers enhanced support for Web Services Description Language (WSDL) with XML Schema Definition (XSD), enabling the use of many industry-standard XML schemas. It also delivers a Business Object Instance Validator with implicit interface-qualifier-based validation and explicit, programmatic validation. It supports Web Services Notification for publish-and-subscribe Web services.

- Seamless integration with the WebSphere platform and currency

  WebSphere Application Server V6.1 allows WebSphere ESB to leverage WebSphere Application Server for other solutions, in addition to inheriting the new features and benefits that were introduced in WebSphere Application Server V6.1.

  It supports diverse sets of platforms, including IBM i, 64-bit Windows® and UNIX platforms, and Windows Vista®. It extends the existing WebSphere MQ messaging foundation to integration new environments in an open, standards-based way. It provides native SCA binding with WebSphere MQ for faster and easier integration. It integrates tightly with IBM Tivoli Access Manager, Tivoli Directory, and Tivoli Composite Application for extending a Federated ESB model.

- Mediation capabilities supporting more sophisticated interactions

  WebSphere ESB has additional primitives, such as a Business Object Map primitive to embed a map inside a mediation flow, a Service Invoke primitive that invokes a target service from within a request or response flow, fan out, fan in, and inline service invocation capabilities, an enhanced Custom Mediation primitive that allows for new terminals to be defined, an enhanced Message Element Setter primitive, and system-wide common database support and schema qualifier on a Message Logger primitive.

IBM WebSphere Message Broker

IBM WebSphere Message Broker V6.1 supports the following key capabilities and benefits:

- Platform-independent Enterprise Service Bus

  WebSphere Message Broker distributes any type of information across and between multiple diverse systems and applications. It reduces point-to-point interconnections and simplifies application programming through separation of integration logic from the applications and from process logic. It has a

powerful publish-and-subscribe matching engine and routes information in real time based on topic and content to any endpoint.

It has a powerful runtime security model, and accelerates Web Services Security processing with DataPower SOA Appliance. It provides integration with WebSphere Service Registry and Repository support to virtualize services for greater reuse.

- Enhance SOA support with advanced Web Services

  WebSphere Message Broker mediates (provides routing, transformation, and logging) between Web Service requesters and providers. It delivers support for Web Services Security and Web Services Addressing, and compliance for Basic Profile. It works with the latest implementations of standards such as Web Services Definition Language (WSDL), Simple Object Access Protocol (SOAP), SOAP with attachments, any Java Message Service (JMS) HTTP(s), MTOM/XOP, and MQ.

- Integration without bounds with connectivity and transformation capabilities

  WebSphere Message Broker has integrated WebSphere MQ transports for Enterprise, Mobile, Real-Time, Multicast, and Telemetry endpoints. It extends the reach, scope, and scale of the enterprise integration bus out to mobile and handheld devices, along with embedded devices such as sensors or actuators.

  It provide native JMS interoperability, acting as a bridge between any combinations of different JMS providers. It contains out-of-the-box nodes to simplify management and improve performance for key enterprise resource planning (ERP) integration scenarios (for example, SAP®, Siebel®, and PeopleSoft®) through WebSphere Adapters packaged as native message-processing nodes. It provides FTP, e-mail, and WebSphere Transformation Extender support.

- Improved consumability, productivity, and systems management

  WebSphere Message Broker allows you to install the product and run a sample in less than an hour. It has simple packaging allowing for easy identification of the appropriate install assets, with a single installation DVD for Windows and Linux desktops. Enterprise-wide identity, authentication, and authorization with Tivoli and Lightweight Directory Access Protocol (LDAP) servers enforce service policy at the user level for greater security.

- Access broader platform support and performance

  WebSphere Message Broker is available on IBM z/OS®, IBM AIX, Linux (System z™, Intel®, and Power), Solaris™ (x86-64 and SPARC), HP/UX (PA-RISC and Itanium™) and Microsoft® Windows Server®. It support databases such as DB2, SQL Server®, Oracle®, Sybase, and Informix®. It supports enhanced SAP IDOC, 64-bit capability on Linux and UNIX, Java Database Connectivity (JDBC™) XA support, and Java 5 on all platforms.

- Enhanced developer productivity

  Message Broker Toolkit is a set of Eclipse-based tools that is compatible with Rational Application Developer V6. It provides data transformation options to match the data format and the developer skill set. It also provides graphical message metadata and mapping definition, visual trace capability, and a message viewer to graphically identify source and target message element locations.

IBM WebSphere DataPower Integration Appliance

IBM WebSphere DataPower Integration Appliance delivers common message transformation, integration, and routing functions in a network device, cutting operational costs and improving performance. By making on demand data integration part of the shared service-oriented architecture (SOA) infrastructure, DataPower is one of the few nondisruptive technologies for application integration.

Features and benefits include:

- Acceleration of existing integration hubs

  Powered by wirespeed transformation technology, the DataPower appliance can be used to off-load XSLT processing, XPath routing, XML conversion, and other resource-intensive tasks from servers to reduce latency, improve throughput, and free up compute resources.

- Mainframe modernization and Web services

  The DataPower appliance can help protect and XML-enable mainframe systems, instantly connecting them to enterprise SOA and Web services.

- Appliance simplicity

  Easy to configure, operate, and own, the DataPower appliance reduces the operational complexity of the IT environment while cutting costs and risks.

- Any-to-any transformation

  The DataPower appliance can parse and transform arbitrary binary, flat text, and XML messages, including EDI, COBOL Copybook, ISO 8583, CSV, ASN.1, and ebXML. Unlike approaches based on custom programming, DataGlue technology uses a fully declarative, metadata-based approach.

- Integrated message level security

  The DataPower appliance includes mature message-level security and access control functionality. Messages can be filtered, validated, encrypted, and signed, helping provide more secure enablement of high-value applications. Supported technologies include WS-Security, WS-Trust, SAML, and LDAP. The DataPower appliance is a sealed network device for high reliability and increased security assurance.

- Sophisticated multi-step message routing, filtering, and processing
- Multiple synchronous and asynchronous transport protocols
- Detailed logging and audit trail

  The DataPower appliance includes non-repudiation support.
- Standards-based interfaces

  The DataPower appliance features standards-based interfaces for broad, uniform connectivity across the enterprise infrastructure.
- Agile, highly flexible underlying scripting/configuration support

  The DataPower appliance enables comprehensive “out-of-the-box” support for fast, easy-to-handle standard, nonstandard, or off-standard, partner/vendor implementations.
- XML enablement and wirespeed application integration

  The DataPower appliance helps reduce the time, cost, risk, and complexity of traditional integration approaches.
- Metadata-based integration

  Unlike approaches based on hardcoded middleware adapters, the DataPower appliance uses a fully declarative, metadata-based approach. The unique technology enables data-oriented programming (DOP) paradigms to be employed without introducing bottlenecks. Easy-to-use visual tools are available to assist business analysts in configuring format descriptors, mappings, and message choreography.
- Security and performance

  Powered by robust technology built from the ground up with security in mind, the DataPower appliance provides XML integration on a more secure foundation with the performance necessary for real-world applications. The DataPower appliance can parse, filter, transform, route, and enforce access-control messages of disparate format at wirespeed. It also provides encryption and decryption, as well as digital signature signing and verification capabilities.

The decision to choose which ESB platform to use is related to the Deploy phase of the SOA life cycle. The criteria for choosing an IBM solution are:

- Use WebSphere ESB when:
  - You use WebSphere Application Server and have skills in WebSphere Administration and Java coding.
  - You are now using or planning on developing a business process using WebSphere Process Server (WebSphere ESB and WebSphere Process Server have a common tooling, programming model, and run time).

- You are integrating with ISV business applications hosted on WebSphere Application Server or third-party solutions that extend and support WebSphere Application Server.
- You are focused on standards based interactions using XML, SOAP, and WS.
- You want to mediate between Web services and existing systems using JMS and WebSphere JCA Adapters.
- Reliability and extensive transactional support are key requirements.
- You want to minimize your server investment by co-hosting WebSphere services and ESB in one application server.

- Use WebSphere Message Broker when:
  - You are currently using WebSphere Message Broker but not as an ESB.
  - You have extensive heterogeneous infrastructures, including both standard and non-standards-based applications, protocols, and data formats.
    - You have extensive MQ skills and infrastructure.
    - You are using Industry formats such as SWIFT, EDI, and HL7.
  - You are implementing a wide range of messaging and integration patterns: complex event processing, message splitting and aggregation.
  - You need extensive pre-built mediation support.
  - You have very complex transformation needs.
  - Reliability and extensive transactional support are key requirements.
  - You want to achieve very high-performance with horizontal and vertical scaling.
- Use WebSphere DataPower when:
  - Ease of use is a pre-dominant consideration. A simple experience of drop-in installation and admin-based configuration with no or minimal development is required.
  - You are transforming between XML-and-XML or XML-and-any other format.
  - Your interaction patterns are relatively simple.
  - Your mediation requirements are met by the existing DP mediations and minimal extensibility is needed.
  - You are using XML-based or WS-Security extensively.
  - You require use of advanced Web services standards.
  - You need to minimize message latency when adding an ESB layer.

- You are doing extensive XML processing combined with high performance requirements.
- Your ESB must be in production very quickly.

WebSphere ESB, WebSphere Message Broker, and WebSphere DataPower can coexist and run within the same infrastructure. Within a federated ESB model, one may choose a particular ESB based upon their specific requirements. Customers can make WebSphere ESB and Message Broker work together using MQ Link, JMS, and Web services (SOAP over HTTP protocols or JMS over either WebSphere MQ or JMS). WebSphere Message Broker and WebSphere DataPower now have a single tool and security policy description for optimal security requirements. With a simple click with WebSphere Message Broker, DataPower will perform WS-Security processing. Applications and message flows will remain unchanged.

An ESB extended

In the fast changing business environment, you need an IT connectivity infrastructure that can handle the requirements of t today as well as position for challenges tomorrow. You need an ESB strategy that will not box you in.

In addition to WebSphere ESB and Message Broker, the following complementary products can further accelerate or extend your ESB implementation:

- WebSphere MQ delivers a proven messaging backbone for SOA connectivity across 80 platform configurations while preserving data integrity end-to-end.
- WebSphere Service Registry and Repository provides an integrated service metadata repository to govern services and manage a service's life cycle. It promotes service visibility, consistency, and decreases service redundancy in your SOA.
- WebSphere DataPower XML Security Gateway XS40 and XML Accelerator XA35 Appliances extend ESB functionality in a hardware form factor with wirespeed performance to simplify, secure, and accelerate your SOA deployment.
- WebSphere Process Server is the scalable, transactional process engine at the heart of your Business Process Management (BPM) and service-oriented architecture (SOA) solutions.
- WebSphere Transformation Extender offers universal transformation deployable anywhere in the enterprise, and provides consistent data transformation and validation across any infrastructure with its enterprise and industry packs.
- WebSphere Adapters helps you to service-enable packaged applications or other assets so they can participate in a SOA.

2.4.3 Role of the WebSphere Service Registry and Repository

The service registry holds a database of published services. It can be thought of as a phone book. It allows consumers to search for a service. The consumer might not know about the service beforehand. The consumer can perform a service discovery, such as looking up all providers that provide a certain service, such as looking for ticket prices.

Figure 2-9 is the traditional diagram used to describe a service-oriented architecture. It consists of a service consumer, a service provider, and a service registry. As SOA is an architectural style, it only defines three roles and the operations they should be able to perform. SOA does *not* define a particular set of underlying technologies to use.

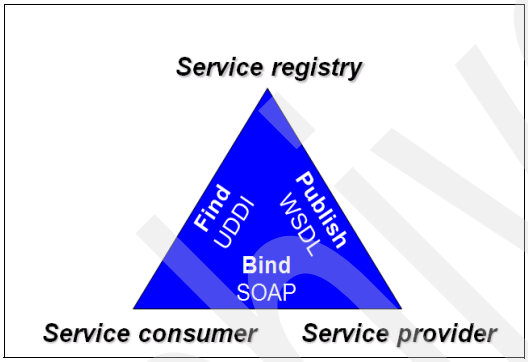


Figure 2-9 Service registry

In Figure 2-9, we add the standard that can be used for the interaction. SOAP, WSDL, and UDDI describes one particular implementation of SOA, that is, an implementation that uses Web services:

- A service registry allows organizations to publish and discover services.
- Businesses can publish (register) a set of services so their internal or external Business Partners can discover them
- Facilitates service discovery, both at design time and at run time.
- Provides registry governance features (management, security, and control of services).

Note: SOA does not require the use of Web service technologies and can be implemented without them. An example of a SOA implemented without Web services would be one based on CORBA or WebSphere Message Broker.

Web services is the leading technology choice for SOA because they are standards based, cross platform and cross language, message oriented, and widely supported. In addition, tooling support for Web services speeds implementation of SOA.

The service registry primary protocol with Web services is the Universal Discovery, Description, Integration (UDDI). An UDDI registry allows organizations to publish and discover Web services. With UDDI, a service provider publishes a set of Web services so their internal or external consumers can discover them.

UDDI is a platform-independent framework for describing services, discovering businesses, and integrating business services using the Internet. Its features include:

- A directory for storing information about Web services
- Support for many types of service descriptions
- The means of forming a relationship between a service provider and a service consumer

The WebSphere Service Registry and Repository (WSRR) acts as the master metadata repository for service descriptions. This uses a broad definition of a service, including:

- Traditional Web services implementing WSDL interfaces with SOAP/HTTP bindings.
- A broad range of SOA services that can be described using WSDL, XSD, and WS-Policy decorations, but might use a range of protocols and be implemented according to a variety of programming models.

As the integration point for service metadata, WSRR establishes a central point for finding and managing service metadata acquired from a number of sources, including service application deployments and other service metadata and endpoint registries and repositories, such as UDDI. Service metadata that is scattered across an enterprise is brought together to provide a single, comprehensive description of a service.

Once that happens, visibility is controlled, versions are managed, proposed changes are analyzed and communicated, usage is monitored, and other parts of the SOA foundation can access service metadata with the confidence that they have found a copy of the record.

In this context, WSRR handles the metadata management aspects of operational services and provides the system a record of these metadata artifacts, that is, the place where anybody looking for a catalogue of all services deployed in or used by the enterprise would go first.

It provides registry functions supporting publication of metadata about services, their capabilities, requirements, and semantics of services that enable service consumers to find services or to analyze their relationships. It also provides repository functions to store, manage, and version service metadata.

It also supports governance of service definitions:

- To control access to service metadata
- To model the life cycle of service artifacts
- To manage promotion of services through phases of their life cycle in various deployment environments
- To perform impact analysis and to socialize changes to the governed service metadata

Business goals for WSRR

Industries are moving to service-oriented architectures (SOAs) to give them increased business agility. Central to this move is the need for a service registry to help manage and govern the services and their business processes.

WSRR addresses this need and the following business goals through service metadata management:

- Business process vitality: Enabling companies to change quickly to address market needs.
  - Rapid selection of different business service providers: Published in the registry and allowing applications to be routed to them.
  - Policy changes to modify behavior: Policies published in the registry together with policy references provided by and required by individual services.
- Business governance: Making sure that the business processes are legal, auditable, and mapped correctly to the IT services.
  - Policy enforcement and change auditing: Providing a simple policy life cycle and status meta data in the registry that can be built on by more sophisticated applications.
  - Security to control who can change policies and the registered service providers: Security-based and role-based access control to meta data managed in the registry.
- IT solution - time to value: Enabling business changes to be brought to the market quickly and at known or lower development cost.
  - Development of new services: The registry ensures reuse of interfaces and existing approved services, promoting interoperability and faster integration.

  - Reconfiguration and assembly of existing services: Building on existing interfaces, protocols, policies, and frameworks and promoting reuse and adoption of standard policies.
- IT governance: Ensuring that the changed systems perform as intended and no other systems are impacted.
  - Service life cycle management: Development, assembly, testing, deployment, operating, versioning, and life cycle status and other metadata held in the registry and used to filter visibility of endpoints according to usage and context.
- Value based QoS: Paying for the quality of service you need, not what you have installed.
  - Policy based service selection: Runtime mediations using policies and metadata managed in the registry.
- Business management: Monitoring what is needed to manage effectively the business and IT solutions.
  - Policy-based logging: Logging and process monitoring defined and controlled by meta data stored in the registry.
- Migration: Moving from traditional architecture to SOA successfully.
  - Discovery and management tools work alongside the registry to help populate it with standards-based mappings from traditional applications.

Using a service registry to improve ESB flexibility

A common ESB mediation pattern is one where mediations are constructed to act as the exposed (or externally provided) services for an organization, as we can see in Figure 2-10.

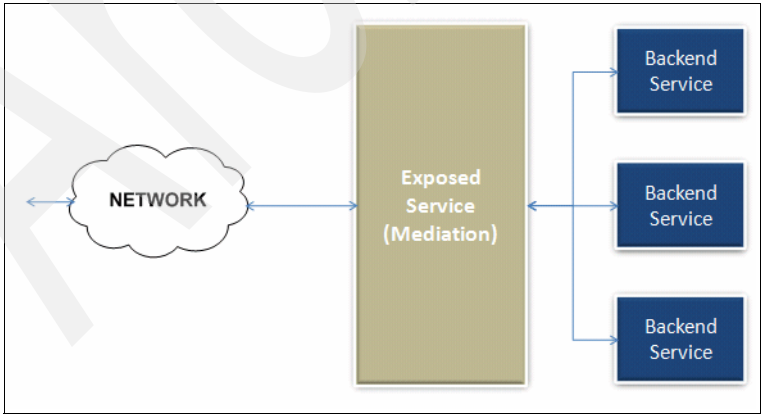


Figure 2-10 Traditional mediation function

The back-end services that the mediation delegates to are generally specified at development time, during message flow development, and usually specify static information about the various service endpoints to be used. As SOA environments become more flexible and responsive, significant numbers of changes can occur within service endpoint definitions (the metadata), making the statically defined mediations brittle. Changing the mediation to accommodate the altered service endpoints raises issues associated with the cost of making a development change, delay in getting these new services deployed, and potential business and operational processes to oversee a mediation change.

Case in point: Consider a mediation that contains the endpoint of the back-end service, such as a Web service, to invoke. This endpoint is expressed using a static URL or other metadata describing a service's binding. This static content makes the mediation brittle when changes are required, such as the deployment of a new Web service, which the mediation now needs to invoke, or deployment of a new back-end service that requires a transformation of the incoming message or that is implemented using a completely different protocol.

You can address the brittleness of the mediation by developing mediations that no longer rely on statically defined information about service endpoints. To do this task, you need a place to get up-to-date information about service endpoints and a mechanism for efficiently accessing that information. Previous attempts at solving this problem have generally relied on some form of access to a data store, such as a relational database or queue. These mechanisms suffered from their proprietary nature, the performance issues associated with database access from a mediation, and the maintenance issues of working with a hand-crafted solution.

A service registry, such as WebSphere Service Registry and Repository, provides a middleware solution to previously hand-crafted mechanisms. You can access up-to-date information about service metadata from Registry and Repository, making the mediations much more immune to changes in service metadata, like service endpoints. For example, using Registry and Repository, you can ensure that the latest release of a particular service endpoint is always invoked, or select a particular endpoint service based on some selection criteria in the input message.

A mediation that uses WebSphere Service Registry and Repository maintains maximum flexibility because the content of the metadata used in decision-making is externalized, as we can see in Figure 2-11 on page 39.

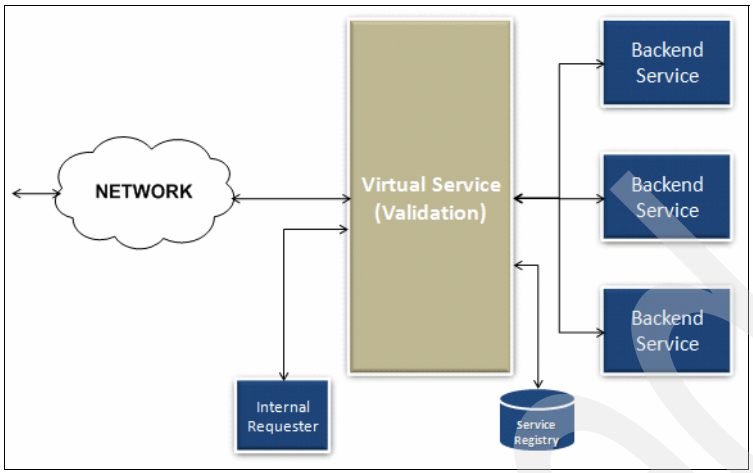


Figure 2-11 Mediation with service registry

An ESB uses a service registry to obtain metadata about the services it needs to invoke. This metadata can be specific service endpoints, supported port types, services that meet a certain SLA requirement, or other metadata. ESB processes the incoming message using a specific method to access the metadata from the service registry. The ESB uses the retrieved metadata to set the correct endpoints to invoke, determine alternate execution paths, enforce policy, or perform other actions.

2.5 IBM SOA Foundation

The IBM SOA Foundation is an integrated set of IBM software, best practices, and patterns designed for enterprises to get started with SOA. It is based on open standards. It views SOA from an architectural perspective. The key elements of the IBM SOA foundation are:

- 2.5.1, “The SOA life cycle” on page 40 helps you understand the SOA dynamics.
- 2.5.2, “SOA solution stack and reference architecture” on page 48 discusses the approaches for implementing SOA.
- 2.5.3, “Logical architecture” on page 54 explains the implementation of several common structures for SOA adoption.
- 2.5.5, “SOA entry points” on page 68 describes some basic options for entering the SOA application fields.

- 2.5.4, “SOA scenarios” on page 61 represent common scenarios of use of IBM products and solutions for SOA engagements.
- 2.5.6, “Use case to scenario mapping” on page 71 maps the scenarios to some real life use cases.

2.5.1 The SOA life cycle

SOA is about a holistic relationship between the business and the IT organization. It contains:

- Tools and methodologies for capturing business design
- Using that design to improve the business
- Tools, a programming model, and techniques for implementing the business design in information systems
- Middleware infrastructure for hosting that implementation
- Management of that implementation to ensure availability to the business
- Ensure efficient use of resources in the execution of that implementation
- Establishment of who has authority
- Processes that are used to control changes in the business design

These activities and the tools involved are best explained in the context of the SOA life cycle.

The IBM SOA Foundation life cycle can guide SOA implementation in creating and integrating solutions and creating a synergy between business processes, business design, and IT. The life cycle phases may vary in their scale and the level of tooling that is used to support each of the steps on its own.

Figure 2-12 on page 41 shows the different stages of the life cycle and the steps involved at each stage. The circle can be compared with iterative software development. Each delivery is followed by another iteration that goes through the same process. Artifacts from previous iterations may be reused for building or incrementing a whole solution.

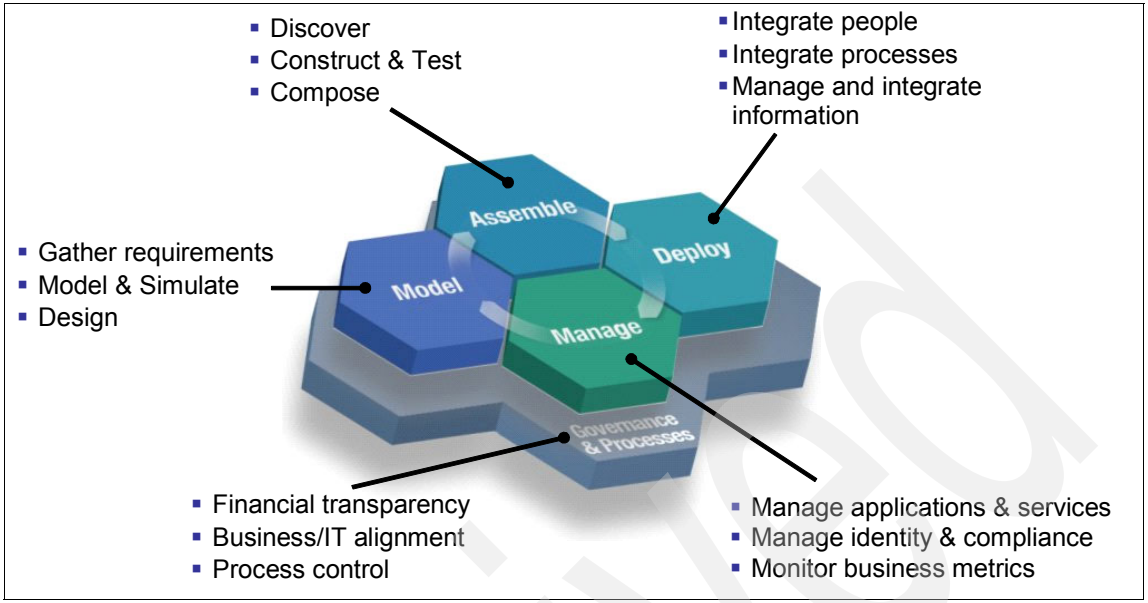


Figure 2-12 IBM SOA Foundation life cycle

Tip: Use the modeling phase extensively. Try to capture differences from the model in other phases and use them as feedback for changes.

Each phase in the life cycle affects other phases, such as:

- Changes to, for example, key performance information in the model phase has to be fed directly to the management phase to align the operational environment.
- Constraints in the Deploy phase, limiting assumptions about where resources are located in the system, may condition some of the Assembly phase decisions.
- Information technology constraints established in the Assembly phase will limit the business design created during the Model phase, for example, the cost of wide-area wireless communication with remote hand held devices may be prohibitive to deploy and therefore needs to be reflected back into the business design.

Effective tool support is crucial when introducing the SOA base on the IBM SOA Foundation. Effective tool usage and support through the life cycle ensures consistency and repeatability of steps and the process. It increases the speed at which the life cycle can be a good modeling tool that can support developers and architects with rapidly generating stubs and adapters for existing components and services.

Each stage corresponds to a number of steps and their results may be actively used in the next step and phase of the life cycle. We describe each step of the life cycle in the following sub sections.

Model

Modeling is the process of capturing the business design from an understanding of business requirements and objectives. The business requirements are translated into a specification of business processes, goals, and assumptions for creating a model of the business and resulting artifacts.

The modeling step can be broken down to the following key activities:

- Gather requirements.
- Model and simulate.
- Design.

Specific tools may support the capture of these requirements in a computerized form that can be used for further modeling steps and transformation of the model to another abstraction level. These tools may support what-if scenarios to simulate changes in the business design. The model should also capture key performance indicators that are important measurements of a business process. The key performance indicators or business metrics should be defined in the design so that they can be measured and used to gain insight in the business health.

Analyze the model performance metrics with a simulation, with the support of the modeling tools (if the are available). This helps identify future performance bottlenecks early and lets you react in a timely manner.

As mentioned, building a good model of the desired architecture and components helps during all phases. The information gathered by the model can then be used to create more detailed information through so called transformations.

The tools for the model phase are:

- IBM Rational RequisitePro®

  IBM Rational RequisitePro is the preferred requirements and use case management tool among practitioners. It is used by analysts to manage requirements and use cases. It integrates Microsoft Word documents and a database to help organize, integrate, and analyze requirements. It displays parent/child traceability relationships, and shows requirements that may be affected by upstream or downstream changes.

- WebSphere Business Modeler

  WebSphere Business Modeler is the IBM offering that allows business analysts to capture business process design. It provides a visual representation of the business processes, organization, resources, and performance metrics. It also includes a simulation tool for process analysis and testing. Finally, the business models can be exported to an IT architect's WebSphere Integration Developer or Rational Software Architect environment.

- Rational Software Architect

  Rational Software Architect is an advanced model driven development tool. It has the capability to turn business models developed in WebSphere Business Modeler into SOA service specifications in UML that are ready for development. It leverages model-driven development with the UML for creating well-architected applications and services. It also supports the application of design patterns, model transformation, reusable asset development, and extensibility.

Some references for the modeling phase are:

- *Model Driven Systems Development with Rational Products*, SG24-7368 illustrates the concepts of modeling and transformations in more detail.
- *Patterns: Model-Driven Development Using IBM Rational Software Architect*, SG24-7105.
- *Patterns: SOA Foundation - Business Process Management Scenario*, SG24-7234.

Assemble

The business design is used to communicate the business objectives to the IT organization. The IT organization assembles the information system to implement the design. The business design is converted into a set of business process definitions. In the business processes, activities are identified to derive the required services. The services get developed into an executable format using transformations of the design or other development work.

The assemble step contains the following key activities:

- Discover.
- Construct and test.
- Compose.

While resolving the design and implementation of the modeled business processes and services, a search of existing artifacts and applications may find components that meet the design requirements and can be reused. Some artifacts fit perfectly, others may be refactored, and some must be augmented to meet the design requirements. These existing assets should be rendered as services for assembly into composite applications. Any new services required by the business design must be created. Service registry can be a good point to start with searching for and registering newly developed services.

The assemble phase includes applying a set of policies and conditions to control how applications operate in the production runtime environment. For example, these policies and conditions include business and government regulations. In addition, the assemble phase includes critical operational characteristics such as packaging deployment artifacts, localization constraints, resource dependency, integrity control, and access protection.

Here are a few tools that can be used in the assemble phase to model business processes for integration, development, testing, and debugging:

- WebSphere Integration Developer

  WebSphere Integration Developer is an Eclipse-based tool, which is designed to help create business process flows, state machines, and business rules. WebSphere Integration Developer has full support for the Business Process Execution Language (BPEL). IBM extensions to BPEL for human tasks add the ability to capture human-to-process, process-to-human, or even human-to-human interaction.

- Rational Application Developer

  Rational Application Developer is an integrated development environment for J2EE programming. Rational Application Developer can provide a transition from modeling, to architecture, to software development, to debugging, and so on. Rational Application Developer covers the spectrum from basic Java programming, to Enterprise Java programming conforming to J2EE, to Portal Server programming. It has a fully integrated debugging tool and a predefined unit test environment for WebSphere Application Server for creating Enterprise Java Bean (EJB™)-based business services.

Deploy

The deploy phase includes a combination of creating the hosting environment for the applications and the deployment tasks of the applications. This includes resolving the application's resource dependencies, operational conditions, capacity requirements, and integrity and access constraints. It allows for the relevant users to access the services through easy to use access methods, may it be through a portlet or a composite application in IBM Lotus® Notes®.

The step of deployment include the following key activities:

- Integrate people.
- Integrate processes.
- Manage and integrate information.

A number of concerns are relevant to construction of the hosting environment, including the presence of the already existing hosting infrastructure supporting applications and pre-existing services. Beyond that, appropriate platform offerings for hosting the user interaction logic, business process flows, business-services, access services, and information logic must be considered. Platform capacity must be considered in anticipation of the additional processing for the new application.

The typical hosting environments and deployment tools used in the deploy phase of the SOA Foundation life cycle are:

- WebSphere Process Server

  WebSphere Process Server is the primary hosting environment for business processing. WebSphere Process Server includes support for both BPEL-based process flows as well as business state machines. WebSphere Process Server also supports the integration of business rules in process and service selection.

- WebSphere Application Server

  WebSphere Application Server serves two roles within the SOA Foundation. It is the hosting environment for basic SOA business services, primarily those implemented with J2EE Enterprise JavaBeans™ components. These services can be exposed with WSDL and integrated through standard Web service protocols and encodings, or can be integrated in a more tightly coupled fashion through Remote Method Invocation/InterORB Protocol (RMI/IIOP) bindings. WebSphere Application Server also serves as the underlying execution platform for WebSphere Portal, WebSphere Process Server, WebSphere Enterprise Service Bus, and a variety of other offerings within the IBM portfolio. This foundational role enables these products to be tightly integrated (albeit hosting a set of loosely-coupled service artifacts) with a common approach to installation, clustering, scaling, administration, service, and security.

The products listed above is just the tip of the iceberg. Depending on the kind of scenario and requirements, IBM has a number of offerings tailored to meet your deployment requirements.

Manage

The manage phase includes the specific tasks, technology, and software used to manage and monitor the services and business processes that are deployed to the production runtime environment.

The manage phase can be broken down in the following key activities:

- Manage applications and services.
- Manage identity and compliance.
- Monitor business metrics.
- Infrastructure management.

Monitoring is a critical part of ensuring the underlying IT systems and application are up and running to maintain service availability requirements and align with service contracts that may have been made in the past to ensure mission critical service availability. Monitoring itself includes these activities:

- Monitoring the performance of service requests and time lines of service responses
- Maintaining problem logs to detect failures in various services and system components, as well as localizing failures and restoring the operational state of the system

Managing the system also involves performing routine maintenance, administering and securing applications, resources, and users, and predicting future capacity growth to ensure that resources are available when the demands of the business warrant using them. The security domain includes such topics as authentication, single sign-on, authorization, federated identity management, and user provisioning. The experiences gathered in such tasks can then be effectively used in future phases of modeling to consider any new constraints.

The manage phase also includes managing the business model, and tuning the operational environment to meet the business objectives expressed in the business design, and measuring success or failure to meet those objectives. SOA is distinguished from other styles of enterprise architecture by its correlation between the business design and the software that implements that design, and its use of policy to express the operational requirements of the business services and processes that codify the business design. The manage phase of the life cycle is directly responsible for ensuring that these policies are being enforced and considered.

Typical tools used in the manage phase of the SOA Foundation life cycle to monitor and manage applications, resources, and business performance indicators are:

- WebSphere Business Monitor

  WebSphere Business Monitor complements WebSphere Business Modeler. It helps create dashboards for visualizing the performance of the business based on the key performance indicators that are identified in the business design (that is, identified in WebSphere Business Modeler and exported to WebSphere Business Monitor). It can be used this to track time, cost, and resources used within the processes. WebSphere Business Monitor provides tools to set situational triggers and notifications to bring attention towards potential bottlenecks or workload imbalances in the business. Ultimately, WebSphere Business Monitor helps you better understand how the business design achieves the business objectives, and provides guidance on how to refine and optimize that business design when the business goals are not being met.

- IBM Tivoli Composite Application Manager suite

  The IBM Tivoli Composite Application Manager (ITCAM) offering is designed specifically to enable IT service management. It has been designed to understand the unique semantics and loosely coupled characteristics of SOA-based services. ITCAM has three editions that are relevant directly to the SOA Foundation: ITCAM for WebSphere, ITCAM for SOA, and ITCAM for Response Time Tracking. These cover application server monitoring and resource consumption, deep-dive diagnostics and correlation as service invocations cascade across multiple systems, and service level response times and problem isolation.

Governance

Governance is critical to the success of all SOA projects. Governance helps clients extend the planned SOA across the enterprise in a controlled manner aligned to the design and architecture of the enterprise. A high level of reusability across the whole enterprise is another result of governance; it drops maintenance costs significantly.

Governance can provide the following benefits:

- Financial transparency
- Business IT alignment
- Process control

SOA governance has four core objectives or challenges:

- Establish decision rights.
- Define high value business services.
- Manage the life cycle of assets with defined processes.
- Measure effectiveness.

IBM SOA Governance and Management Method (SGMM) is a complete process for performing the SOA governance life cycle. It helps identify the appropriate best practices, merged with your existing IT processes, to provide proper governance of the capabilities introduced with SOA. The end result is a project plan to create your organization's unique governance framework.

2.5.2 SOA solution stack and reference architecture

In this section, we introduce SOA reference architecture, also referred to as the SOA solution stack. The SOA solution stack provides an architectural template for architecting solutions. The template is based on the building blocks of SOA: services, components, and flows that collectively support business processes and goals. It provides a high-level abstraction of a SOA factored into layers, each of which addresses specific value propositions within SOA. Underlying this layered architecture is a metamodel consisting of layers, architectural building blocks (ABBs), relations between ABBs and layers, interaction patterns, options, and architectural decisions. These guide the architect in the creation of the architecture. In other words, the SOA solution stack helps answer the question: “If I build a SOA, what would it conceptually look like and what abstractions should be present?”

Reference architecture layers

Figure 2-13 on page 49 depicts the SOA reference architecture from a solutions point of view. The SOA solution stack has nine layers that are designed to reinforce SOA business value. There are five horizontal layers, in the SOA solution stack, relate to the overall functionality of the SOA solution. The four vertical layers are non-functional in nature and support various concerns that cut across the functional layers.

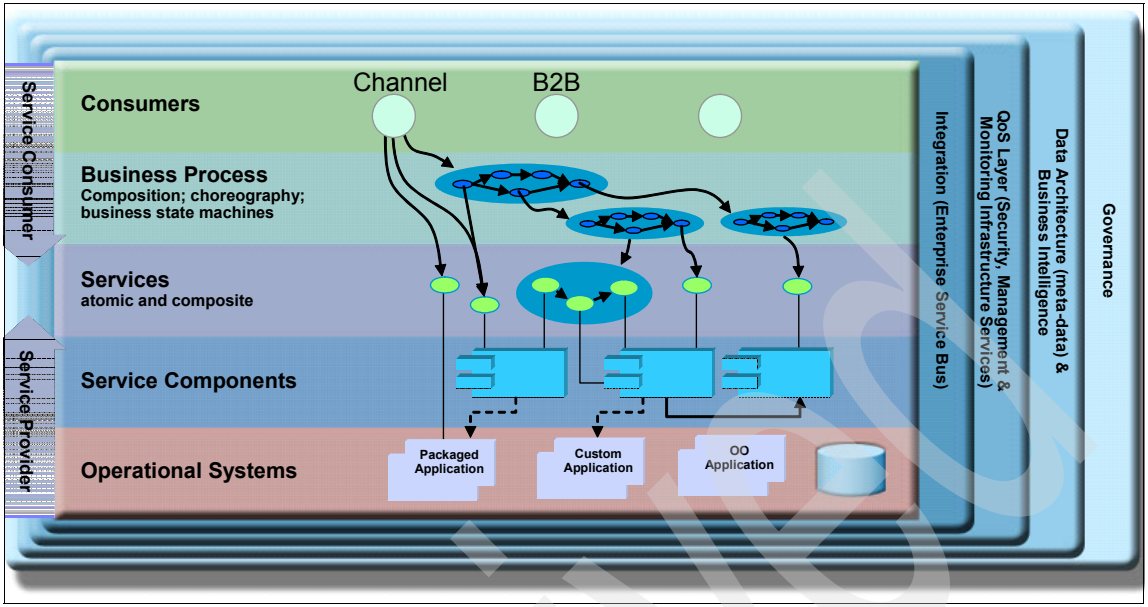


Figure 2-13 SOA Foundation reference architecture: solutions view

Each layer has logical and physical aspects. The logical aspect includes all the architectural building blocks, design decisions, options, key performance indicators, and so on. The physical aspect of each layer covers the realization of each logical aspect using technology and products.

The functional layers are:

- Operational systems layer

  This layer includes all application assets in the application portfolio for the enterprise. It includes all portfolios that are supporting business activities to some extent. The operational layer is made up of existing application software systems; it is used to leverage existing IT investments in implementing a SOA solution.

- Service component layer

  This layer contains software components that provide the implementation and realization of services. Service components reflect the definition of a service, both in its functionality and its quality of service. The service component layer should conform to service contracts defined in the services layer. It guarantees the alignment of the IT implementation with the service description.

- Services layer

  This layer consists of all the services defined within SOA. A service is considered to be an abstract specification of business-aligned IT functions. The specification provides service consumers with sufficient detail to invoke the business functions exposed by a provider of the service.

  Exposed services reside in this layer; they can be discovered and invoked or possibly choreographed to create a composite service. The service layer also takes enterprise-scale components, business-unit-specific components, and project-specific components and externalizes a subset of their interfaces in the form of service descriptions. Thus, the components provide services through their interfaces. The interfaces are exported as service descriptions in this layer, where services exist in isolation (atomic) or as composite services.

- Business process layer

  The business process layer covers the process representation, composition methods, and building blocks for aggregating loosely coupled services as a sequenced process aligned with business goals. Information flow is used to enable interactions between services and business processes. The interaction may exist within an enterprise or across multiple enterprises.

  This layer includes information exchange flow between participants (individual users and business entities), resources, and processes in a variety of forms to achieve the business goal. Most of the exchanged information may also include non-structured and non-transactional messages. Business logic is used to form service flows as parallel tasks or sequential tasks based on business rules, policies, and other business requirements. The layer also includes information about data flows within the enterprise or across multiple enterprises.

- Consumer layer

  The consumer layer, or the presentation layer, provides the capabilities required to deliver IT functions and data to users to meet specific usage preferences. This layer can also provide an interface for application to application communication. The consumer layer of the SOA solution stack provides the capability to quickly create the front end of business processes and composite applications to respond to changes in user needs through channels, portals, rich clients, and other mechanisms. It enables channel-independent access to those business processes supported by various application and platforms. It is important to note that SOA decouples the user interface from the components. Some recent standards such as Web Services for Remote Portlets (WSRP) Version 2.0 can be used to leverage Web services at the application interface or presentation level.

The vertical layers are:

- Integration layer

  The integration layer is a key enabler for a SOA because it provides the capability to mediate, route, and transport service requests from the service requester to the correct service provider. This layer enables the integration of services through the introduction of a reliable set of capabilities. These include modest point-to-point capabilities for tightly coupled endpoint integration as well as more intelligent routing, protocol mediation, and other transformation mechanisms often provided by an enterprise service bus (ESB). Web Services Description Language (WSDL) specifies a binding, which implies the location where a service is provided. An ESB, on the other hand, provides a location-independent mechanism for integration.

  The integration that occurs here is primarily the integration of layers two through four. These are the layers that provide communications, invocation, and quality of service between adjacent layers in a SOA. For example, these layers are where binding of services occurs for process execution, allowing a service to be exposed consistently across multiple customer-facing channels, such as Web, IVR, and the like. The transformation of response to HTML (for Web), Voice XML (for IVR), or XML string (for Siebel client) can be done using XSLT functionality supported through ESB transformation capability in the integration layer.

- Quality of service layer

  The QoS layer provides a SOA with the capabilities required to realize nonfunctional requirements (NFR). It must also capture, monitor, log, and signal noncompliance with those requirements relating to the relevant service qualities associated with each SOA layer. This layer serves as an observer of the other layers and can emit signals or events when a noncompliance condition is detected or, preferably, when a noncompliance condition is anticipated.

- Information architecture and business intelligence layer

  The information architecture and business intelligence layer ensures the inclusion of key considerations pertaining to data architecture and information architectures that can also be used as the basis for the creation of business intelligence through data marts and data warehouses. This includes metadata content, which is stored in this layer, as well as information architecture and business intelligence considerations.

- Governance layer

  The governance layer covers all aspects of business operational life-cycle management in SOA. It provides guidance and policies for making decisions about a SOA and managing all aspects of a SOA solution, including capacity, performance, security, and monitoring. It enables SOA governance services to be fully integrated by emphasizing the operational life cycle management aspect of the SOA. This layer can be applied to all the other layers in the SOA solution stack. Because it helps enforce QoS and makes the appropriate application of performance metrics, it is well connected with the quality of service layer.

  This layer can speed the SOA solution planning and design process. The governance layer provides an extensible and flexible SOA governance framework. This provides guidelines for solution-level and service-level agreements based on QoS and KPIs, a set of capacity planning and performance management policies to design and tune SOA solutions, and solution-level security enablement guidelines from a federated applications perspective. The architectural decisions in this layer are encapsulated in consulting practices, frameworks, architectural artifacts, documentation of SOA capacity planning, any SOA-solution SLAs, SOA performance-monitoring policies, and SOA solution-level security-enablement guidelines.

  The following Web site provides more information about governance:

  `http://www-306.ibm.com/software/solutions/soa/gov/method/`

This pattern of layering based on functionality and purpose may be seen as familiar as it realizes a clear separation between presentation logic, business logic, and data logic as introduced by three tier distributed application architectures and the Model-View-Controller pattern. Considering Business Application Services as a decomposition of Process Services will guide you towards a n-tier distributed system architecture emerging from the SOA Foundation. This is no accident. The basic principles that argue for a separation of concerns between these aspects of an application design still hold within a service-oriented architecture and, as such, are explicitly exposed in the architecture. Service orientation allows you to keep this structure more easily than usual applications as long as the services themselves do not mix different layers in their operation.

Architecture implementation

SOA creates a premium for the role of the enterprise architect who is responsible for modeling and decomposing the business design to align it to physical assets and future services, with intent to reuse and openness for future reuse by undefined components. A major task during this task is to find a common and major base for reuse of the actual services that are then being used to access the underlying information provider.

Usually, there are different approaches for starting with the reference architecture and the modeling of the business design. We speak about top down, bottom up, and meet in the middle approaches, where each one has a different benefit that has to be taken into account.

- Top down approach

  When taking a top down approach, the enterprise architect starts by identifying the business processes and business services used by the business consumers on the top layer. The business users are consumers of the processes and services in the level below. Business processes should be treated as compositions of other business processes and services, and therefore should be decomposed into their subordinate sub-processes and services.

  Services and business processes are then detailed in service components that provide related functionality. Service components include a detailed set of definition metadata used to describe the service to the information system. Services can be aggregated into module assemblies. The module assemblies are used to establish related design concerns and begin the planning that determines what teams will collaborate to implement the related services to be deployed as a single unit. The resulting set of business process definitions, services, and schemes make up the logical architecture of the application. The enterprise architect must then map this logical structure to a physical architecture or existing assets.

  As this approach starts with identifying the business related layers first, a benefit is a very business efficient SOA. However, if we refer to the challenge of integrating SOA into an existing enterprise, the manageability of existing data is a major concern that is addressed late in the top down approach. If data is associated across different systems, this can increase the burden of accessing this information effectively through services.

- Bottom up approach

  In this approach, the enterprise architect starts with the underlying layers. He identifies possible service components used to access data and business related information in an effective and reusable way.

Based on the data that is accessible, these components can then be used to address a higher set of business processes and service compositions that will result in a set of defined services. Other services that are needed to further address a specific business issue are then added on demand with a possible reuse of existing, isolated (atomic) services.

Given the source of service creation in the underlying data structure, this approach seems best suited when we talk about data integration, and not focusing on addressing business issues and providing business flexibility, as this is achieved late in this approach. It is comparable with the modernization of older assets across the enterprise.

- Meet in the middle approach

  The meet in the middle approach is an approach that unifies the two other approaches and tries to find common ground. Therefore, it is the mostly widely used approach of the three.

  In the meet in the middle approach, the services and service compositions are modeled with the business process, business consumers, and infrastructure with its data and information in mind. Given the reference architecture shown in Figure 2-13 on page 49, the meet in the middle approach can be seen as a process that is applied with all the layers being vertical and creating the solution by designing and developing horizontally to those layers.

  This approach achieves a good way to meet business requirements and data integration across the life cycle. The logical structure is mapped to the physical architecture at the same time the business consumers are being identified and addressed.

2.5.3 Logical architecture

The SOA reference architecture model attempts to decompose the functional underpinnings of your application design. Figure 2-14 on page 55 illustrates the logical architecture. This architecture helps maintain a clear separation of concerns. The separation enables us to focus attention on the special skills that are required for each section of the overall architecture, enabling resource optimization of the skills required for a given topic. This specialization avoids a situation where everyone on the team needs to understand everything about the entire system to be effective at anything they do with part of it. This should lower the cost of training, enable more efficient implementations, and enable the construction of tools optimized for specific skill sets.

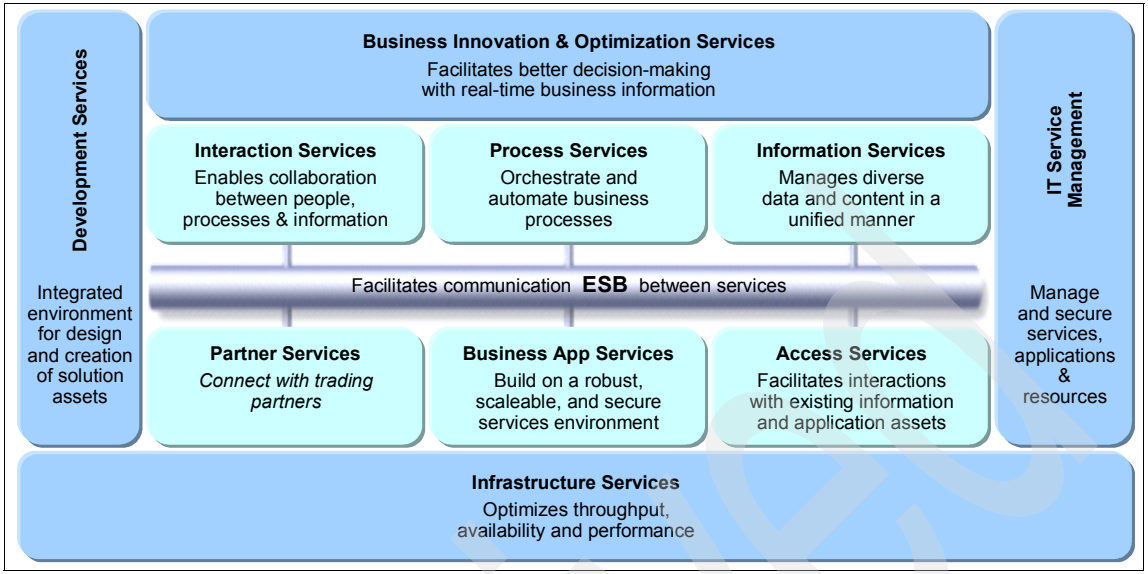


Figure 2-14 SOA Foundation reference architecture: logical architecture

The SOA logical reference architecture attempts to be comprehensive enough to cover all aspects of an enterprise.

The light blue highlighted boxes are the parts in which the application software that captures the domain logic of a business design will be deployed. These parts usually communicate through the middleware layer and are realized by an Enterprise Service Bus. The other service areas exist to provide various supporting functions in the SOA life cycle. They are used in the modeling of the business design, construction, and assembly of the software, deployment of the applications, and management of operational systems and optimization of the business design that is implemented.

Interaction services

The area of interaction services can be seen as being responsible for the presentation logic of the business design, that is, what is used for interacting with the business, and either receives or provides information. The interaction focuses on the communication between the actual user and the business application. We are referring to an user not only as a person, but also other endpoints, such as robots, sensors, and so on. A sensor may provide, for example, feedback to the business logic that is awaiting a decision about whether the temperature is still within the limits for storing the products properly or not.

Those interaction services may be very specific to the role they are tailored to, for example, a robot for has different requirements for receiving a table of data than a human does. Profiling the information effectively and in the most generic way so that it can be reused across different roles is one challenge in the area of interaction services. Always keep in mind that the goal is to maximize the output or input capabilities based on the role and the user involved in the interaction.

Interaction services will be primarily used in the consumers layer, but it can also be used on the lower layers of the solutions view in Figure 2-13 on page 49.

Process services

The area of process services can be seen as an abstraction towards the business design. They provide the required control capabilities for managing the flow and interactions of multiple services in ways that implement business processes. They include various forms of compositional logic, for example, business process flows, finite state machines for business composition, and so on. There are also other forms, such as business rules, decision trees, and dynamic compositions to provide a choreographed service composition. Depending on the business design, one may be better suited than another.

Business rule engines are also one possible way to create an abstraction from the actual business of the process. It can, for example, abstract checking if an item is taxable for the given geography and the amount of tax to be levied out of the business process. Hence, the change of the tax information because of regulations is isolated from the actual business process. This also provides loose coupling inside the processes; a domain can operate and decide independently. The business rule engine is responsible for consulting the appropriate source for the decision that has to be made, in our example, a tax table.

Process services operate primarily for the business processes layer when referring to the solutions view in Figure 2-13 on page 49.

Business application services

Business application services are the core of the business logic that have been decomposed down to the flexibility needed in the business design. These are service components created specifically as services within a business model and that represent the basic building blocks of your business design. Each service on its own may not have any effect or provide any results, but as part of the process services, for example, they realize their power.

These services may also be invoked directly from the presentation logic, for example, to get the shelf where an item is stored.

Recalling the solutions view of the reference architecture in Figure 2-13 on page 49, the invocation of the business application services is primarily for the service components and services layer itself.

Information services

The area of information services can be separated into the areas of master data management, business intelligence, content management, information integration, and data management.

Services in this area contain the actual logic of the business design at two levels. They provide access to the persisted data of the business in, for example, query statements for retrieving the information or integrity checks for the data manipulated by services. These data services are mainly available to the business application services and aligned with a specific domain model so as to reflect the business entities.

The information services have the goal of creating a logical view towards the data, including a federation across different data sources with simple access for consumers. For example, a logical view to a single row may include access to different data sources such as files, tables in a database, and Web resources.

Another requirement addressed by the information services sub-architecture is the movement from one part of the enterprise to another as needed to satisfy its own data flow and life cycle requirements.

Access services

The area of access services can be seen as a proxy to provide access and utilize older applications and functions to the SOA components. This is usually realized by applying an adapter pattern to the component and then making this adapter accessible as a service. (See *Design Patterns: Elements of Reusable Object-Oriented Software*, by Gamma, et al for more information about the adapter pattern.)

For most of the existing applications in an enterprise, this might be achievable by utilizing either developed access points in the application or by hooking into extension points. For other proprietary applications, it might be necessary to access provided APIs in order to gain access to their underlying logic and expose them as a service.

Access services can be bound to different levels of the solutions view, as some adapters may just expose data on the service components level, while others may provide adapters to existing business processes.

Partner services

Partner services capture the semantics of partner interoperability that have a direct representation in the business design. This can, for example, include policies and constraints that other businesses must conform to in order to work with the business, including business requirements, such as the need to conform to specific industry messages and interchange standards and protocols. It can involve the business logic of managing how partners are selected, and which ones are used as a preference over others in particular circumstances.

In some ways, partner services are somewhat analogous to interaction services, that is, projecting a view of the business to partners, and controlling the interaction with them as external entities. In other respects, partner services are analogous to access services, that is, rendering the capabilities of that partner as a service so that those functions can be composed into the business processes like any other service.

Partner services operate at the same levels as the solutions view in Figure 2-13 on page 49 as interaction services and access services.

Enterprise service bus

The enterprise service bus we introduced in 2.4.2, “Enterprise Service Bus and SOA” on page 22 is a silent partner in the SOA logical architecture. Its presence in the architecture is transparent to the service consumer and service provider of a SOA application. However, the presence of an ESB is fundamental to simplifying the task of invoking services, that is, making the use of services wherever they are needed, independent of the details of locating those services and transporting service requests across the network to invoke those services wherever they reside within the enterprise or in which departmental domain they are managed.

Access services that wrap existing applications may exist wherever they have been deployed: on different servers, in different departments, or in different data centers. The enterprise service bus simplifies the task of incorporating these disparate systems so that they can be exploited in the business design.

The enterprise service bus can be found as a surrounding component in the solutions view. From a services view, the enterprise service bus can be seen as component that bridges the service consumer and service provider.

Business innovation and optimization services

These services primarily represent the structures for encoding the business design in a flexible way, including business policies and objectives. Business innovation and optimization is achieved by capturing the business design and then focusing on that design to improve it through a combination of iterative refinement and analysis of real-time business metrics.

Business innovation and optimization services exist in the architecture to help capture, encode, analyze, and iteratively refine the business design. The services also include tools to help simulate the business design (see the IBM SOA Foundation life cycle step “Model” on page 42) and to use these results to predict the effect that design, or changes to that design, will have on the business.

The services will be linked directly into the information system (a particular component in the IT service management box) to both collect performance metrics coming out of the system as well as to change which metrics are measured as monitoring needs change.

Like the enterprise service bus, these services can be seen as the surrounding box in the solutions view in Figure 2-13 on page 49. This may vary from the business goals as, for example, a help desk solution may have other optimizations in mind than a hosting solution for large distributed databases.

Development services

Development services encompass the entire suite of architecture tools, development tools, visual composition tools, assembly tools, methodologies, debugging aids, instrumentation tools, asset repositories, discovery agents, and publishing mechanisms needed to construct a SOA based application.

Using tools that can also communicate together and exchange information, you achieve the development with perfect tool support across all the involved roles. The transformations in the life cycle step, discussed in “Model” on page 42, can be effectively used throughout the development to share data in different forms. A good platform for such tools is the Eclipse framework that allows easy plug-in services. Eclipse is used by IBM tooling support to support service orientation.

Like the enterprise service bus and business innovation and optimization services, development services can be seen as encompassing the SOA solution and all its layers. Whether it is development of a logical data view or the development of XML transformations for the data representation to the user, tooling support can decrease the development time and support different roles of developers or engineers.

IT service management

Once the application has been deployed to the information system, it and the IT infrastructure on which it is hosted need to be managed. IT service management represents the set of management tools used to monitor service flows, the health of the underlying system or databases, the utilization of resources, the identification of outages and bottlenecks, the attainment of service goals, the enforcement of administrative policies, and recovery from failures.

Since the business design is established and captured as a model, it is possible to capture correlations between the business and the IT system. This correlation, if carried into the deployment environment, can be used by IT service management services to help prioritize the resolution of problems that surface in the information system, or to direct the allocation of execution capacity to different parts of the system based on service-level goals that have been set against the business design. Basically, this can be seen as a feedback between the model, business design, and the deployed solution.

As with the previous services, IT service management is an area applied across all layers of the solution.

Infrastructure services

Infrastructure services form the core of the information technology environment for hosting SOA applications and the layers involved, including middleware components. It is through these services that it is possible to build a reliable system to provide efficient utilization of resources, ensure the integrity of the operational environment, balance the workload to meet service level objectives, isolate work to avoid interference, perform maintenance, secure access to confidential business processes and data, and simplify overall administration of the system.

Infrastructure services may even be able to virtualize the underlying computing platform and resource dependencies. These services are built using SOA principles, exploiting the characteristics of loose coupling to enable highly flexible and composable systems. This allows you to assemble a set of capacities and capabilities that are needed to achieve particular business needs.

We will delve deeper into the topic of how beneficial actual system functionalities and features for the infrastructure services can be in Chapter 4, “IBM System p features” on page 99.

Infrastructure services can be seen as a carrying layer for the whole solution, that is, encompassing it in the solutions view. The promotion of features and benefits the infrastructure provides to higher layers illustrates the importance and benefit of these services in the IBM SOA Foundation.

2.5.4 SOA scenarios

The SOA life cycle helps you to organize a project plan around the four phases that constitute them. However, it would be very convenient to develop common and recurring themes and patterns for the application of SOA-based solutions and identify common situations in which SOA is being used today. IBM has eight such SOA common usage patterns. These patterns are collectively called the SOA scenarios.

IBM introduced the SOA foundation scenarios to ease the way towards the SOA implementation details. The SOA scenarios quickly communicate the business value, architecture, and IBM open standards-based software used within a scenario. The scenarios can be implemented as part of an incremental adoption of SOA growing from one scenario to using elements of multiple scenarios together. The concept of realizations are used to provide more specific solution patterns and IBM product mappings within the SOA scenarios.

Figure 2-15 on page 62 shows the scenarios and how these scenarios relate to each other. The scenarios may be used together to support an entire solution. The relationships shown may not be the only set of ways in which the scenarios may be interrelated. When more SOA experiences are gained, additional well-proven relationships may be added.

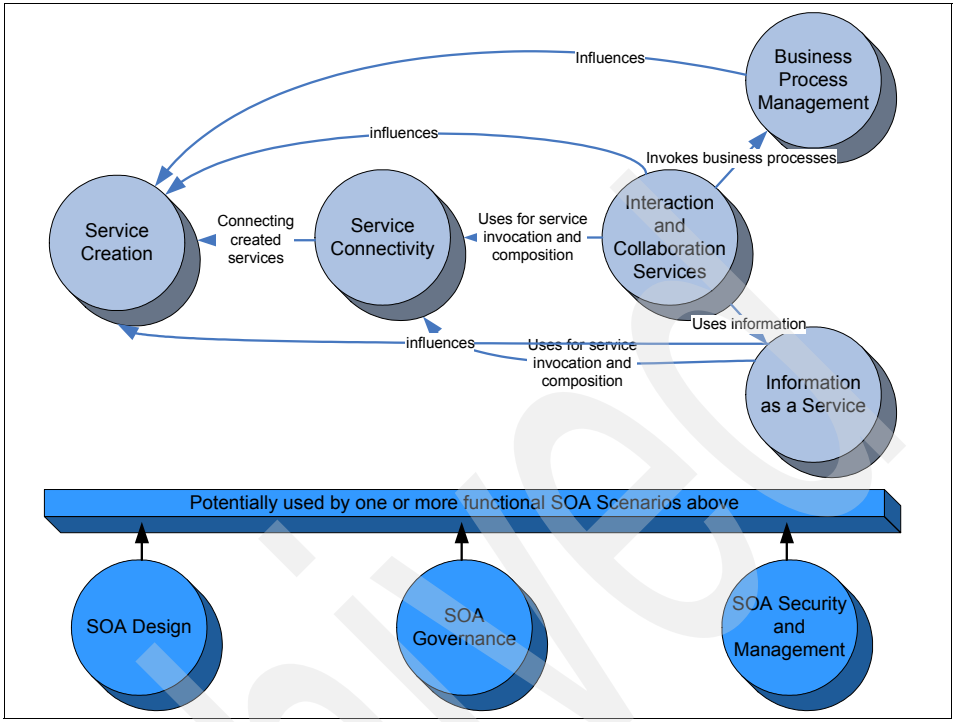


Figure 2-15 Common scenario relations

Based on customer experiences and years of input, IBM realized that there were several common scenarios businesses tended to follow on their way to designing, implementing, and integrating SOA solutions in the enterprise. By defining these scenarios, IBM provides predefined, real world approaches for implementing SOA solutions. Each scenario provides tested and integrated product offerings, or realizations, that can then be used to implement the scenario. So the scenarios can be mapped to a company's specific goals and needs and gives you a good idea of how it is possible to realize these benefits.

To select an appropriate scenario based on the customer requirements, use the process in Figure 2-16 on page 63. The first step is capturing the customer requirements, business context, IT context, and the customer's vision of the future business and IT environment.

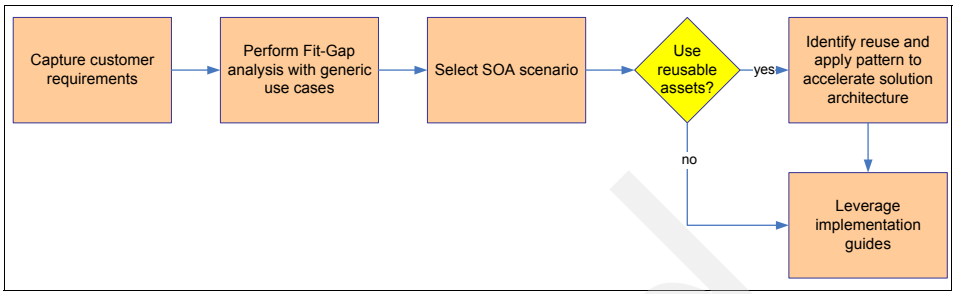


Figure 2-16 Scenario selection process

The SOA scenario framework lists generic use cases drawn from experiences analyzing customer requirements across many industry sectors. Each real use case identified in the first step is then analyzed against the list of generic use cases to evaluate which generic use case maps closest to the customer use case. We discuss these use cases in 2.5.6, “Use case to scenario mapping” on page 71.

Service creation

This scenario is about creating flexible, service-based business applications. A new service-oriented application exposes business behavior as a service and also reuses business logic that is exposed as a service. This scenario maps directly to the entry point Reuse, as it exposes business behavior and reuses existing business logic.

There are many possible examples to illustrate the service creation scenario in an enterprise. We shortly describe four implementations that illustrate key elements of the service creation scenario:

- Directly expose existing applications as services.

  This realization directly exposes existing applications as services. The existing application can be a wide range of application types, such as an EIS (for example, CICS® or IMS™), a J2EE application, SAP, and so on. A key distinction for this realization is that the service interface to be exposed is defined by the existing application, so a direct proxy rather than a new logical view or adapter that utilizes multiple functionalities is provided over one interface.

  This implementation refers to the “Bottom up approach” on page 53, as the underlying structure is being used to expose interfaces.

- Indirectly expose existing applications with service components.

  For this implementation, the existing application functionality is exposed indirectly using service components. In this realization, business alignment is achieved by defining the service interface independently of existing assets. This approach to creating a service refers to the “Top down approach” on page 53. In practice, the implementation of the service may be a combination of top-down and meet in the middle.

  When using the techniques found in Service-Oriented Modeling and Architecture (SOMA), top-down analysis is used to define a service independently of existing assets. This approach helps to define better what a service should do or how it should look, based on business domain analysis and decomposition. A second phase of service identification called *existing asset analysis* is used to identify opportunities for reuse that might exist among existing programs and so forth.

  Service components provide the linkage between the domain-driven service definition and the reuse of existing assets. The service component interface is derived from the business driven service description while the implementation of the service component takes advantage of any existing assets that have been identified for reuse. The implementation typically aggregates existing behavior using adapters or connectors. This usually involves some refactoring of the existing assets. In comparison with directly exposing assets, this approach can be compared with the adapter pattern, where a logic view based on upper layer requirements is used for the interface creation.

- Create a service from an interface.

  In this realization, a new service is created from an existing service interface. This realization can be described as a “create from scratch” approach, as everything is based on the abstract interface and no underlying assets may be appropriate for reuse. This realization uses a top-down approach from the interface on towards creating the service.

- Consume services from external service providers.

  In this realization, client applications consume services from external service providers. This realization represents the view of a consumer that is using one or more external services. The service consumer only sees the service interfaces and uses them to compose a business flow that implements the use case. The endpoints and any security or transport constraints are imposed by the service provider.

  For example, a client application may invoke an address verification service from a third-party service provider to implement the use case of listing verified addresses where invoices have been sent to.

The realizations are representative of the service architecture and the method in which the service is created. The service architecture describes whether the application functionality will be exposed directly as a service, or indirectly using a middle-tier service component. The realization examples include different approaches to creating services such as bottom-up and top-down. There is a wide range of existing application types such as EIS (for example, CICS and IMS), J2EE, SAP, and so on.

More about this scenario can be found in *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240.

Service connectivity

This scenario is about linking people, processes, and information in the business together with a seamless flow of messages and information. The origin is from virtually anywhere at anytime by making a set of core services available to a variety of applications through the usage of an intermediary service gateway or bus. This scenario directly maps to the entry point Connectivity.

The focus of the scenario is on the underlying connectivity used to support business-centric SOA. An enterprise service bus provides de-coupling between clients and providers to create the flexibility to implement applications more quickly. In circumstances where services are provided to or consumed from a third party, an ESB gateway can be used in conjunction with the ESB to add security measures, as shown in Figure 2-17.

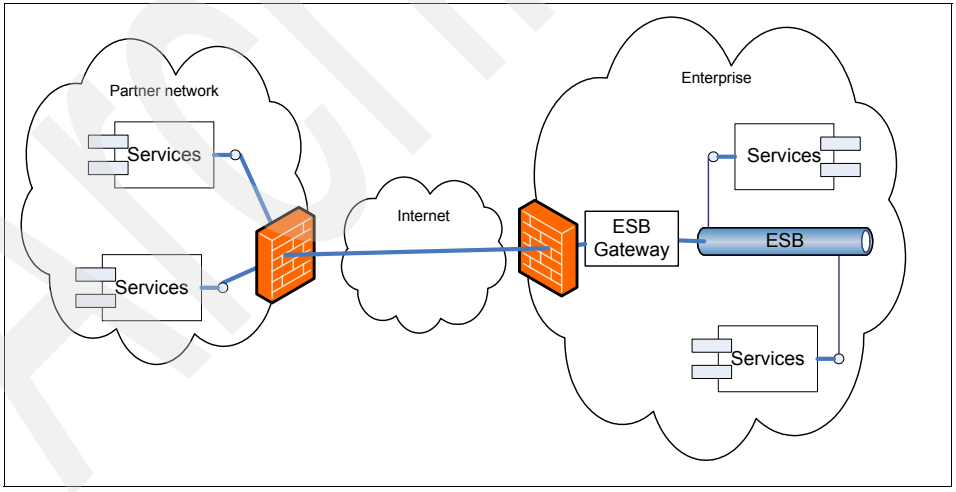


Figure 2-17 ESB gateway for accessing services beyond enterprise boundaries

Implementations of this scenario have the following features:

- Enables changes to the implementation of a service without affecting clients.
- Registers services to a service registry.
- Uses an enterprise service bus as the integration point between service providers and consumers.
- Enables clients to access a service with a different interface and protocol than what the service consumer supports.
- Uses an ESB gateway to isolate and protect services.
- Enables management and monitoring of services to insure service level agreements.
- Provides security and credential mapping (where needed) to insure proper use of the services.

More information about this scenario can be found in *Patterns: SOA Foundation Service Connectivity Scenario*, SG24-7228

Interaction and collaboration services

This scenario hooks in when a service or set of services must be presented to a human user through multiple devices, such as a browser, PC, and mobile devices. Interaction and collaboration services also improve people productivity by aggregating these services as views delivering information and interaction in the context of a business process and specific to the roles of people. This scenario corresponds to the entry point People.

For an example of this scenario, refer to *Case Study: Interaction and Collaboration Services SOA Scenario*, REDP-4375

Business process management enabled by SOA

Business process management is a discipline combining software capabilities and business expertise to accelerate process improvement and facilitate business innovation. This scenario refers to the entry point Process and facilitates the composition of services across the enterprise to achieve business results. The main area of this scenario can be seen in the realization of “Process services” on page 56.

More about an implementation of this scenario can be found in *Patterns: SOA Foundation - Business Process Management Scenario*, SG24-7234.

Information as service

The information as a service scenario provides access to complex, heterogeneous data sources within the enterprise as reusable services. This scenario refers to the entry point Information and includes the previous introduced topics of data federation and creating an accessible service logic to present a clear and consistent view and access to data.

As introduced in “Information services” on page 57, this includes master data management, business intelligence, content management, information integration, and data management.

For a practical example of this scenario, refer to *Case Study: Information as a Service SOA Scenario*, REDP-4382.

SOA design

The SOA design scenario is about aligning the modeling of business design and IT solution design through a set of roles, methods, and artifacts to render a set of explicit business processes for optimization and services for composition and integration. This scenario does not refer to any specific entry point; it applies to all of them and can be seen as supporting their realization.

The realization of a SOA design based on an example is discussed in *Case Study: SOA Design Scenario*, REDP-4379.

SOA governance

This scenario establishes and enforces SOA development and runtime processes. It defines the policies, processes, and tools that oversee who owns a service, who can use it, how it can be used, and when it will be available to those needing it. As a SOA design, this scenario corresponds to all entry points.

Governance based on an example is discussed in *Case Study: SOA Governance Scenario*, REDP-4384.

SOA security and management

This scenario includes application of security and management policies within the realm of SOA. Security encompasses the need for a security model that enables secure business transactions across and between enterprises. The security domain includes such topics as Web Services Security, authentication, single sign-on, authorization, identity propagation, federated identity management, and user provisioning.

Management includes solutions for managing and monitoring composite applications. It also includes discovery, monitoring, securing, provisioning, change, and life cycle management of services as a part of IT Service Management (ITSM). As SOA design and SOA governance, this scenario belongs to all entry points.

Case Study: SOA Security and Management Scenario, REDP-4378 describes the security and management based on an example and helps in understanding the actual role of it in an implementation.

2.5.5 SOA entry points

IBM organizes the SOA scenarios around five distinct yet interrelated entry points that help enterprises get started and succeed with SOA. An enterprise can get started with an approach that begins with the fundamental assets of the enterprise: people, information, and processes. It can also lay the technical groundwork for integration by leveraging the entry points of Connectivity and Reuse. Regardless of the approach, the SOA entry points can help the business pursue SOA at a pace that suits its needs. The entry points describe starting points across all different roles.

Business driven entry points

The business driven entry points relate to people, information, and processes.

People

The people entry point is a starting point for SOA to enable people to interact with application and information services that support business processes. As a complement to the other entry points (process, information, reuse, and connectivity), the people entry point can facilitate real time decision making and dynamic collaboration, and immediate execution. Overall, the people entry point approach to SOA drives business and operational flexibility and improves user productivity and collaboration.

This entry point refers to the consumer layer discussed in 2.5.2, “SOA solution stack and reference architecture” on page 48, which allows profiled interaction with defined users. It utilizes the area of “Interaction services” on page 55 to specifically provide a service abstraction for the representation to the user.

Employee productivity, operational efficiency, and the ability to innovate on the fly based on information and business design are significant to competitiveness and growth. Companies frequently struggle with domain specific applications and information that prevent customers, employees, and partners from working together effectively and utilize a common base of information. Because people drive the interaction with the SOA services that execute business results,

focusing on people is critical to the success of SOA implementations. The people entry point to SOA can help to:

- Accelerate productivity
- Reduce costs for access to multiple applications and information sources
- Reduce time to deploy for new services
- Increase access to process flexibility and orchestration
- Enable collaboration inside and outside the enterprise
- Provide role based content easily

Process

The process entry point is a business centered starting point for service-oriented architecture (SOA) that provides specific tools and services to help streamline and improve processes across the enterprise.

This allows processes to quickly respond to changing market conditions and business design changes. By streamlining the processes, it is possible to align the business and IT goals and reduce the complexity of building new processes. SOA, with a focus on processes, will:

- Improve the efficiency through more business centered interaction.
- Increase collaboration through easy to access processes.
- Accelerate the time to market through easy to create processes.
- Respond quickly to business challenges.
- Implement new processes in less time.

The entry point focuses on the area of “Process services” and “Business application services” on page 56. In reference to 2.5.2, “SOA solution stack and reference architecture” on page 48, these services can be found primarily on the levels of business processes and services layers.

Information

Information as a service is an entry point to SOA that offers information access to complex, heterogeneous data sources within an enterprise as reusable services. These services may be available both within the enterprise and beyond.

This entry point corresponds to the scenario discussed in “Information as service” on page 67. It can be seen as the underlying logical view to the data other services may access. Exposing information towards services provides the following benefits:

- Data accessibility:
  - Develop a unified view of the business with access to analytical data for improved transparency and business insight.
  - Generate and govern master data records with shared metadata.

- Reduce cost and risk:
  - Reduce costs associated with infrastructure rationalization and migration by decoupling information from specific information sources.
  - Reduce risk exposure through analytics and auditable data quality.
- Increase agility:

  Increase the agility for business transformation by providing reusable information services, spanning structured and unstructured information that can be plugged into applications.

IT driven entry points

The IT driven entry points relates to connectivity and reuse of the services.

Connectivity

Service connectivity is an IT-centered entry point to SOA designed to simplify the IT environment with a more secure, reliable, and scalable way to connect within and beyond the enterprise. The connectivity refers not only to the infrastructure, it allows, on higher levels, linking people, processes, and information of the business environment into a seamless flow of messages and information from virtually anywhere at anytime. SOA brings new levels of flexibility to such linkages in other ways than loose coupling.

One of the major contributions to connectivity can be seen in service bus services, which are discussed in “Enterprise service bus” on page 58, which takes over most parts of the connectivity between services and therefore business design. Additional areas like data federation can be seen as parts of connectivity.

Service connectivity will help to:

- Ensure a seamless flow of information.
- Execute broad business processes that span across the enterprise and beyond.
- Scale the enterprise so that it grows smoothly.
- Delivers a consistent user experience regardless of the transportation medium

Reuse

Service creation and reuse is an IT-centered entry point to SOA that focuses on deriving continued value from previous assets. By entering from the service creation and reuse entry point, existing systems and services will be captured in the design and modeled towards effective usage in a new service-oriented architecture to be used enterprise wide and beyond. This allows new exposure of

existing assets that are able to interconnect to other services rather than having a domain specific or proprietary limited effect to the business.

The IBM entry strategy to SOA through service creation and reuse can help:

- Reduce the amount of new code that must be created.
- Reduce maintenance costs by eliminating redundant systems.
- Create composite functions by incorporating existing applications.
- Integrate tasks performed by existing applications.

2.5.6 Use case to scenario mapping

The IBM SOA foundation contains some use cases. Use cases are practical usages of a scenario, with a complete analysis of the components and their implementation. Table 2-1 shows the relation of generic use cases to entry points.

Table 2-1 SOA use cases

Generic use case	Reuse	Connectivity	People	Process	Information
U1: Reuse existing or create new application logic as a service within the enterprise.	X				
U2: Reuse existing or create new application logic as a service outside the enterprise.	X				
U3: Point-to-point integration of enterprise applications using services.	X				
U4: Point-to-point integration of intra-enterprise applications using services.	X				
U5: Allow users to invoke services simply.	X				
U6: Enable loose coupling of service consumers and providers using static routing.		X			
U7: Enable loose coupling of service consumers and providers using dynamic routing based on standards-based protocols.		X			
U8: Enable loose coupling of service consumers and providers using advanced dynamic routing and diverse protocols.		X			
U9: Improve an existing business process flow through business process and policy modeling and simulation.				X	
U10: Implement a new business process flow.				X	
U11: Analyze existing business process flows using monitoring.				X	
U12: Allow single sign-on access to different services.			X		

Generic use case	Reuse	Connectivity	People	Process	Information
U13: Personalize information based on user profile.			X		
U14: Allow users to create and manage content.			X		
U15: Allow users to access services through client devices.		X	X		
U16: Allow users to perform information inquiries.					X
U17: Populate information.					X
U18: Allow users seamless access to diverse data sources.					X

The SOA design, SOA governance, and SOA security and management scenarios are applied across all use cases based on the particular requirements of an implementation.

3

SOA infrastructure selection

This chapter highlights the impact of SOA on the IT infrastructure. We try to capture infrastructure aspects of a SOA that are different from the traditional architectures. Through this chapter, we explore the architectural principles, guidelines, and other considerations for deciding on an infrastructure for your SOA-environment. The discussion is divided into:

- 3.1, “Mapping the infrastructure” on page 74
- 3.2, “SOA requirements on infrastructure” on page 74
- 3.3, “Key elements of a SOA infrastructure” on page 77
- 3.4, “Architecting SOA infrastructure” on page 87
- 3.6, “Conclusion” on page 97

3.1 Mapping the infrastructure

Historically, IT infrastructure was dedicated to application silos. Application infrastructure is allocated in a static way. Often, each line of business has their own sets of servers that support their own sets of applications. IT had a limited view of its function to the business, as it is more concerned about assets. With SOA, IT has a direct alignment to the business. It is not so much about managing specific assets or looking at the IT components as it is about how IT supports the business and how IT creates value by aligning the business goals and enabling business innovation. The infrastructure and IT enables the business to do what the business needs to do to succeed.

The world of SOA changes the way that businesses view IT and how IT views businesses. Its also redefines a whole new outlook at various aspects of the IT realm. In particular, it gives the world of IT infrastructure a whole new meaning. IT infrastructure, in the SOA world, no longer means the just the physical infrastructure. It has now become the confluence of middleware and hardware that enables IT to help businesses realize their investments.

It is worth pointing out at this juncture that the move to SOA has encompassed the IBM middleware programming model as well as lower level infrastructure software, for example, system management and storage management application programming interfaces and functions. IBM Server and Technology Group products incrementally expose management infrastructure through a SOA model. The Open Grid Services Architecture (OGSA) and Web Services Distributed Management (WSDM) are examples of evolving infrastructure standards and management standards based on Web services.

3.2 SOA requirements on infrastructure

SOA applications require the same end-to-end performance, security, management, availability, quality of service, and so on. How does SOA impact the infrastructure and management of the infrastructure?

Apart for the standard design requirements, there are differences of in SOA application design. Traditional application infrastructures are designed in a monolithic way. A well-defined stack of an application load and a non-configurable application structure allows a sizing estimate to be quite reliable. Also, SOA based applications are dynamic. New applications can be constructed based on existing services by adding new components, thus adding unforeseen loads and structures.

It is more difficult to understand the application profile because of the complexity of the changing environment.

- How do you manage performance when demands can change at any time?
- How do you manage availability when the end-to-end application path may be redesigned tomorrow?
- How do you predict the demand for services when new business process may use the service anytime?

Services can be reused. One business unit may create a service that may be leveraged by another business unit or another application in the future. That can potentially and very quickly change the performance characteristics and the demand on that service. These changes must be accommodated.

Services are combined from multiple sources. Distributed and composite applications add new twists to managing the quality of service. You must be able to see and control the service level that is affected by a myriad of components underneath. For example, a composite service may be composed of services from various departments and communicates over a bus. In this case, you need to take care of many aspects, including the bus, application, and various other connecting service and infrastructural aspects to ensure proper operation of the components. In some ways, SOA is meant to simplify the applications, but if you are not careful, the infrastructure can actually be made more complicated.

Services can be deployed rapidly. That is one of the reasons that businesses are interested in SOA. They want to be able to adapt the business quickly to the changing business climate and they can do so by assembling the services in different ways that already exist. Through reuse, they can quickly reduce the time to market for a new application. This has a significant impact on things like Release Management and Change Management. These become very difficult topics to manage in a service-oriented environment because of the need for quick deployment. They also, as mentioned earlier, can affect the predictability of demand and performance. If you do not understand how the new organization of the application is going to work, you will not be able to understand the demand and the performance impact on the system.

Services route to any available resource, so SOA is about having virtualized resources both at an application level as well as at an infrastructure level. So when something executes in the runtime environment today, that runtime environment may be located in a different portion of your network or in a different data center tomorrow. Potentially, some of the execution could be offloaded or outsourced to a service provider providing that service externally. This has a profound impact on the performance of the overall application end-to-end. The distributed access again provides broadening access to information and applications. This also provides greater security risks. As the small components

that make up the application are located throughout the network and outside of your network, you have to be concerned about access to those services in a very different way.

In principle, an existing infrastructure design can be tailored to support new Web services requirements. We are not talking about dramatically different technologies here. This is really the same old infrastructure. We have servers, networks, and middleware software. So, in principle, you could tailor the infrastructure towards the new Web services. But the difference in infrastructure between SOA and Web services is more about providing management, security, performance, availability, and scalability to support services, not applications. So, you really have to break that paradigm and think about very fine-grained services or small applications, not monolithic applications. And although these considerations are applied today to traditional system design, they can be applied to SOA if you take the additional view towards services. Also, the design methods in reference architectures that most of us use today are still applicable to building robust and mature infrastructures to support Web services and SOA. The biggest difference really is the need for the infrastructure architect to obtain information about the application. (Not that this was not done in the past, but often IT organizations only made a cursory understanding of what the application really did.) Today, it is absolutely critical that you fully understand the application and how that application is going to use the infrastructure so that you can properly architect the infrastructure.

Here are some key questions that must be answered when designing an infrastructure for SOA:

- Infrastructure and services security
  - How do we secure services?
  - How does this affect my overall infrastructure security goals?
- Systems performance
  - How will XML transformation between our older and distributed systems, affect application performance?
- Reliability Availability and Serviceability
  - What happens if one of my services is unavailable?
  - Where are my applications located?
  - What are my dependencies between my Business Partners?
  - How do I restore and resynchronize any associated data and metadata to keep data consistency?
- Scalability
  - How do I ensure that the infrastructure will grow in line with volume?

- Manageability
  - How do I manage my services to tell whether they are available and performing?
  - How do I validate my applications and ensure that they are meeting business goals?

3.3 Key elements of a SOA infrastructure

Figure 3-1 highlights the key elements of a SOA Infrastructure. It is quite similar to an e-business Web-based application infrastructure. There are DMZs and different security zones. There are servers, portal servers, and application servers. However, there are additional components that are very important to SOA and only relevant within a SOA environment.

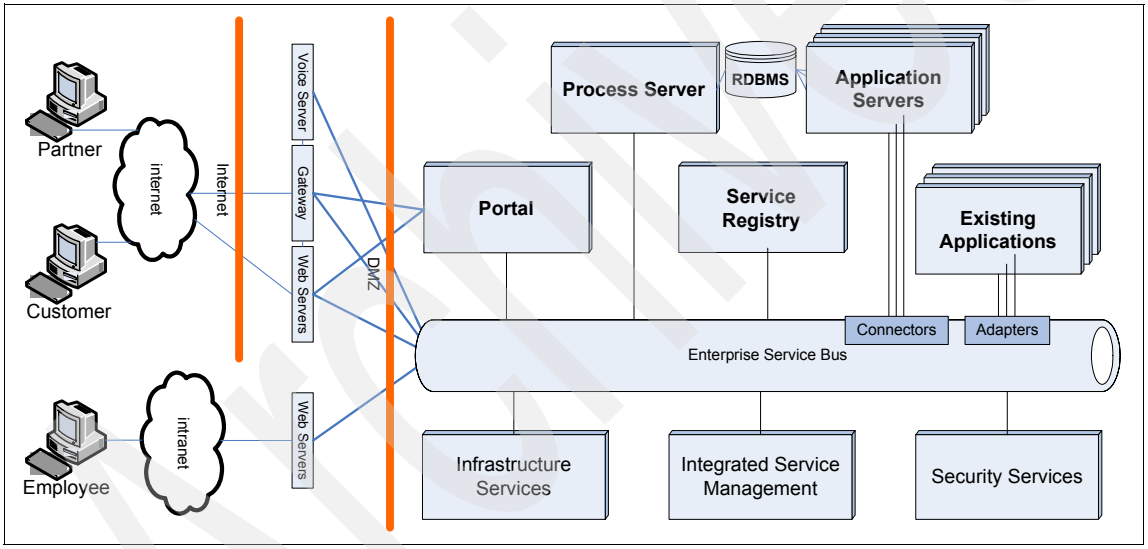


Figure 3-1 Key elements of a SOA infrastructure

In the center of Figure 3-1, there is the Enterprise Service Bus. The ESB (discussed in 2.4.2, “Enterprise Service Bus and SOA” on page 22) is an infrastructure that can transport messages between application components. This is a very important concept in SOA because it allows decoupling of applications. You do not have the direct application to application communication that has caused many problems.

Enterprise Service Bus acts as the layer to which each application can communicate. Enterprise Service Bus can make decisions on where to route the messages and what transformation of the message content needs to be made. The messaging routing and transformation is handled in this layer instead of in the application. This makes modifying or changing applications in the future much easier. It makes the communication between services much easier.

One of the challenges for ESB is how you find the service to which you need to connect. This task is done through the service registry. The service registry acts as a repository of the services that are available and describes their interfaces. The service registry can be queried for a location of the appropriate service. This is so that you can drive reuse and understand what is available to the application owner.

SOA applications are made up of many small applications or services. SOA needs to orchestrate or coordinate the execution of these various services. This function is performed by the process server.

These components (ESB, service registry, and process server) are integral parts of your overall solution. In the SOA world, the overall solution really applies to the entire enterprise. Many applications are going to be using these services. These components are going to have very high security requirements and very high availability and performance requirements.

With SOA, the Enterprise Service Bus is very critical. Should it become unavailable or become impacted by performance issues, it could potentially impact all of your applications and affect your business processes. It is absolutely critical that these components be architected in a way that can meet the levels of service that are required by the business.

3.3.1 Mediation and registry considerations

The ESB solution must be simple enough to meet your existing requirements. At the very least, ESB must provide some value by transporting messages. The ESB solution must also be able to scale. It must have the capability to add more complex routing and transformation of messages. Complex ESB may need to be federated. Figure 3-2 on page 79 shows the various capabilities of an ESB.

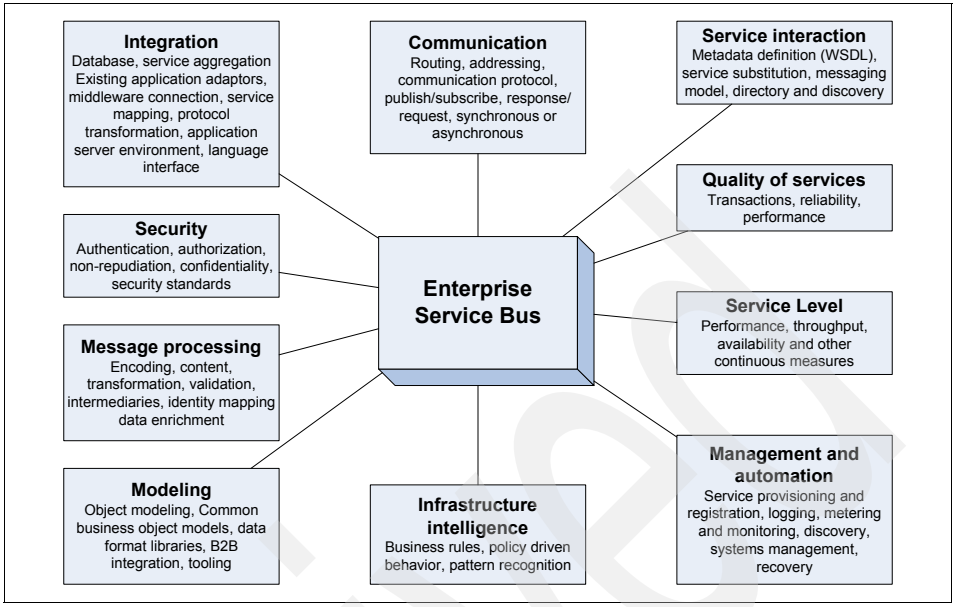


Figure 3-2 Capabilities of an ESB

The Enterprise Service Bus should provide basic capabilities such as communication and integration. It also allows security and message processing to be enforced. It also facilitates additional capabilities ranging from service modeling, service interaction, service provisioning, and service level management. It may also involve management and automation, infrastructure intelligence, and ensuring quality of service.

The important concept that needs to be recognized is that the ESB has separated infrastructure services from application and business services. Building logic for routing or transformation into the application adds unnecessary complications to the application and can be best handled in the services of the Enterprise Service Bus. The main idea for an ESB capability is clear separation of concerns based on the premise of *keeping infrastructure services separated from business services*.

A logical services model is an essential component of a SOA design. It helps to specify the proper separation of concerns within the ESB. This can be understood by revisiting the SOA principles. To truly achieve SOA, the Enterprise Service Bus must separate service consumers and providers from each other to abstract them from each other so that services can be switched or changed.

To summarize, ESB allows externalize protocol transformation and routing. It keep infrastructure services separated from business services. SOA design must leverage the capabilities of the ESB and Service Registry. The Service Registry provides a common repository for asset sharing.

3.3.2 Security

SOA security is one of the large changes and challenges in the infrastructure design. Security has been a large concern in monolithic applications from a more holistic standpoint. In a SOA environment, on which components are loosely coupled and distributed, the communication security and application security requirements multiplies exponentially. With the alignment of IT systems to business processes, security practices have to be adapted and aligned to the business processes as well.

Figure 3-3 identifies some important security concerns in a typical infrastructure.

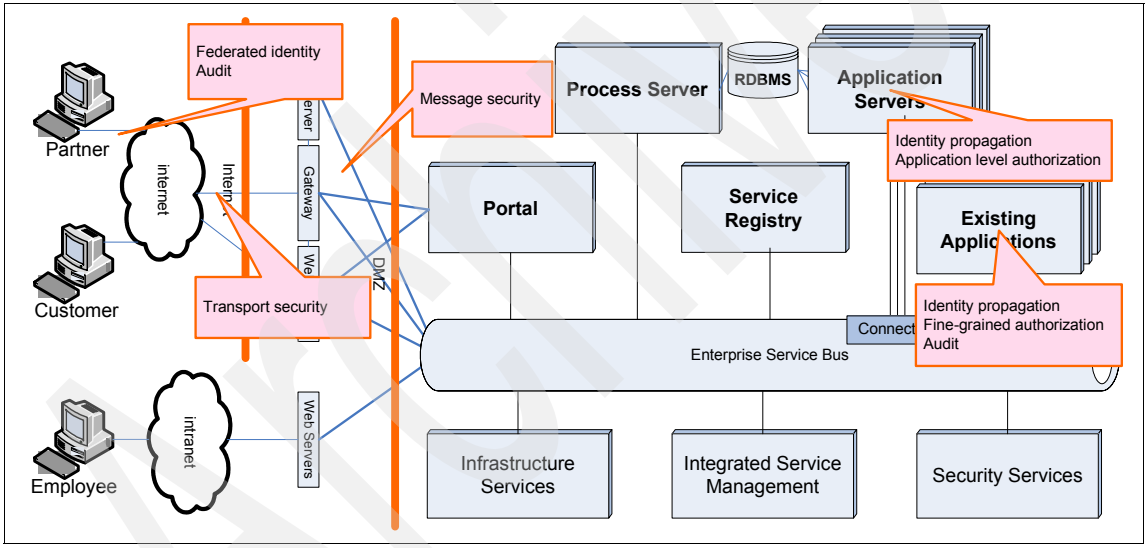


Figure 3-3 Key elements of a SOA infrastructure

Service orientation aims to provide services that are interconnected and reused as appropriate to fill particular business processes. These services must be connected and implemented in a secure and auditable manner according to a defined security policy. Therefore, identity and security policy play key roles in architecting SOA solution. Identities exist for both users and services and both should be subject to the same security control. Security policies define the requirements for access to be provided for services. These identities may need to be propagated through the SOA environment. In many cases, service

implementations may restrict the options and formats available for propagation of a user's identity.

Identity services are therefore required in the infrastructure, not the application. The infrastructure and mediation must handle identity issues so that services can easily be interconnected without worrying about how to map or propagate a user's identity from one service to the next.

Role-based access controls for services are used to reduce user administration. Users are assigned to roles. Each role has a certain authorization to access the services. User membership to a particular role enables access to a service rather than explicitly linking access to individual user identify.

The following are some additional security considerations:

- Connecting to other organizations on a real-time transaction basis

  There is a need to connect to services that exist outside of your organizational boundary. These inter-organizational interactions may occur in a SOA deployment. This can involve integrating the service user interfaces from different domains or organizations into a single portal interface. Another example may involve a service provided by a partner organization being directly invoked from a business process within your organization. Regardless of the form of the interaction, it is imperative that the security, identity, and access policies are defined and enforced for all of these transactions. These policies must be enforced for both incoming and outgoing requests. Secure boundary services or firewalls are obvious starting points. These boundaries, secure services, should be able to provide coarse-grained verification that requests are coming from or going to trust parties. But you still need to establish the trust relationships between these organizations and that is the key to inter-organizational cooperation. So, you must set up some rules, as federating the information is a very complex concept. IBM Tivoli products can assist you with this task.

- Security in composite applications

  Services may require specific privileges from the invoker to access the service. However, when a service is combined, such as when you are choreographing them in a business process, the combination of these services may require another examination of the security access policy. You may define an additional policy or role that is not the same as the individual services access. This can lead to additional administration processing.

- Managing identity and security across diverse environments

  A typical SOA environment has many points that identify the security policy and enforce it. These policy enforcement points are located both at the service connectivity level as well as within the implementations of the services themselves. A challenge is management of the policy across these various heterogeneous enforcement points. This is difficult because the management and administration interface may not be the same. Integration with an existing application on a different platform is another issue. Distributed security management is not advisable, as discrepancies may generate security holes that lead to exposure. Protection of data from unauthorized modification and disclosure is key with SOA.

- Demonstrate compliance with a growing set of standards

  There are many standards and regulations that require the ability to show compliance. Auditing of transactions is very critical. We are not in a monolithic application where auditing can be performed from certain points in the application. Auditing has to be performed across multiple tiers. We must have an ability or service to monitor and provide that compliance, and that is often done through some sort of auditing process where access is logged and audited at each point of the SOA environment.

The following guidance applies to designing security in a SOA environment:

- Security authorization needs to be granular at the service level.
- The requirements drive the design.
- Understand the existence of corporate security policies, and use descriptive over programmatic approaches to allow decisions to be set at deployment time.
- Evaluate performance impacts.
- Security is obviously very important, but you do need to understand and balance the impact on performance.

3.3.3 Performance

A performance model should be created and maintained throughout the project life cycle as the application is built. This is not a new requirement, but it is very critical in the SOA world. Distributed application performance, in which components exist in multiple servers with different monitoring and measurement schemes, can be difficult to achieve.

To establish a performance base line, you must measure, manage, and adjust performance throughout the cycle of the project. Performance testing must obtain enough consistent metrics in order for you to gain confidence in the results and

validate that the solution meet performance expectations. You must perform tests multiple times, get multiple data sets, and get comfortable with the information to verify that you are going to be able to achieve your performance expectations.

The clustering technology of the application server and service provider implementation that allows a new process to be provisioned in the case of a performance need is beneficial. Monitoring and automatic actions resolving the performance problem in a transactional perspective is necessary. Identifying a bottleneck in a single application platform does help, but it is no longer adequate. Distributed workload management and monitoring can also assist in defining the solution.

3.3.4 Infrastructure management

The ultimate success of SOA lies in the area of infrastructure management. You want to ensure that your enterprise system management frameworks are extended to include these new SOA applications. Without the ability to effectively manage and monitor these infrastructure services and business processes in an integrated view, you will not have the necessary insight into your environment to support SOA. SOA and Web services management is an extension of enterprise service management, which most businesses have deployed within their data centers. SOA management tools can hook into your existing systems management framework and help provide insight.

SOA is about simplifying applications to be more agile and to adapt to the changing business climate. However, it may make the infrastructure so complex that you lose visibility into the infrastructure. This results in significant problems with quality of service and the ability to deliver reasonable performance for the application. You may not achieve the goals of SOA.

Figure 3-4 shows the various layers of the SOA solution stack over which monitoring and management capability would need to be enabled in a typical infrastructure.

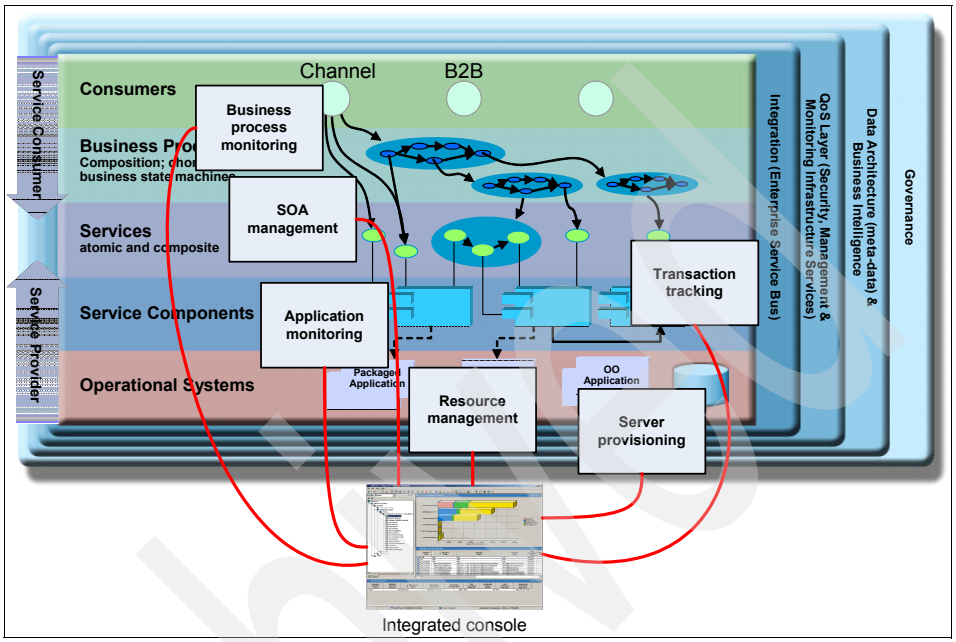


Figure 3-4 Service management and monitoring

Figure 3-4 shows the IBM Tivoli Monitoring (ITM) and IBM Tivoli Composite Application Manager (ITCAM) solution. ITCAM has agents in the SOA environment that allows you to manage and monitor components of your SOA environment. With the integrated console, that is, the Tivoli Enterprise Portal, you can get an end-to-end perspective of your SOA environment. It can alert you to potential problems and even has automatic actions. This is very critical because these environments can become complex.

A monitoring agent on just the portal server or an ESB server is not enough. You must have all the layers of the component monitored so that you can be aware of failure at all levels. Are all your power supplies available? Is the server up? Is it connected to the network? Is the network transmitting packets? You need to be able to manage all these levels and be able to observe them from a SOA standpoint in order to manage them.

One of the big differences about managing a SOA environment is you are not just managing a vertical slice or a monolithic application. You manage an infrastructure vertically, but you are also managing horizontally across the

business process, so you need to determine if the business process is complete. This happens across many, many infrastructure components.

You must automate, provision, and control the services to meet the service level. In a SOA environment, the service use will change. As new uses for the services are identified, they quickly get used and the performance indicator changes potentially for the application. The performance analysis metric must be modified and capacity planning must be adjusted. This can be a major issue if you cannot react as quickly as your application grows.

You can also make use of tools to discover relationships and improve application availability. Obviously, you can see in a complex environment such as this one that the topology of the actual application can be very difficult. Better tools to discover and map out that topology are starting to appear. Because it is such a dynamic environment, topology changes happen on a day-to-day basis. It is no longer enough to have a chart that says this is how the transaction works, this is the servers it runs on, and so on. That may change from day-to-day or hour-to-hour. So you need automated tools to show the current topology at the time of the problem. To achieve a good SOA implementation, you must be successful at general infrastructure management.

3.3.5 Infrastructure virtualization

Virtualization in the general infrastructure has been prominent for a few years now, and it is also now a concept in the application architecture of SOA. This section attempts to link some of the synergies between the two. Horizontal integration dictates breaking down the silos of the application. In SOA, we talk about leveraging processes, business processes, and services and reusing those services in multiple lines of business and not looking at them from a siloed standpoint.

When we architect virtualization solutions in the infrastructure, we are also looking at it from a very similar perspective. We are trying to break down the siloed infrastructure, standardizing it, and providing common resources that can be used by the different servers across the enterprise.

Why are we doing SOA? We are doing it to be more agile, which supports the business. We want to be quicker. We want to be able to adapt our business and, therefore, the application to meet the needs of the business. And on the infrastructure side, we have a very similar value that we can provide in virtualization. Using virtualization technologies, we can provision partitions or environments very quickly and bring those online to meet changing business requirements. Enabling flexibility in SOA supports dynamic application construction. If something needs to change, you can often change the order of

the services that are called or add a service that potentially adds a new function to that business process.

Virtualization can happen on many levels, such as the processor or server level (as with VMware), on the storage level with Storage Area Network (SAN), and the whole infrastructure, such as logical partitioning. Complex interdependencies are removed in monolithic applications for well-documented business process flows that are managed and orchestrated through the process server on the SOA application side. On the virtualization side of the infrastructure, we simplify the infrastructure but, by not tying specific hardware levels and specific servers to a specific application, we gain flexibility and simplification by abstracting it through a virtualization layer.

SOA needs dynamic resources, because it is all about dynamic applications being orchestrated and combined in different ways. We have the ability with virtualization to be able to support SOA on a virtualized environment. In fact, virtualization will also benefit from SOA. It is easier to virtualize a SOA environment because of the nature of the fine-grained services that can be deployed into the virtualized server resources. So again, as the usage patterns of these services change, we can adapt the virtualization technologies more quickly and have a flexible and adaptive IT infrastructure.

Some guidance on virtualization:

- The first step is to simplify by consolidating servers and reducing the footprint. At least from a physical hardware standpoint, that simplifies the environment, simplifies the storage, and moves away from attached storage towards consolidated storage area networks. This reduces the complexity of the environment.
- Start to virtualize. IT resources are used across applications without regard to where they physically reside. So we break the notion of an application running on server X and now have the application run within this pool of resources, which makes better use of the infrastructure.
- Automate and provision your environment. Building environments in the past were error-prone because they were done manually. By automating things you can get it done quicker and consistently.
- The ultimate goal is full orchestration or coordination of your resources, dynamically allocating the capacity to meet business goals for the increased infrastructure, agility, and readiness for growth.

3.4 Architecting SOA infrastructure

The preceding sections talked about the various considerations to be taken into account when architecting an infrastructure for your SOA. In this section, we will attempt to build a high level series of steps that can be used to arrive at the right SOA infrastructure.

3.4.1 Design stages

The design stages that we discuss in this section are:

1. "Requirements capture" on page 87
2. "Solution analysis and design phase" on page 87
3. "Map components to the SOA reference architecture" on page 90
4. "Document architecture decisions for integration" on page 90
5. "Software and middleware product mapping" on page 91
6. "Define the Logical Operational Model" on page 92
7. "Identify the deployment pattern for each product" on page 93
8. "Evaluate physical platforms" on page 93
9. "Capacity planning" on page 94
10. "Resource planning" on page 94
11. "Resource management" on page 95

Requirements capture

The first step, as with any project, is to capture the requirements to the finest level of detail. The requirements include the business case, business vision, pain points, and success criteria, along with the use cases, such as usage scenarios to be addressed, to deliver to the business the values and benefits it is looking for. From an Information Technology stand point, capture the various non-functional requirements (such as performance, availability, serviceability, and scalability) that would need to be taken care of. Also look at any specific standards or compliances that need to be followed or addressed. The existing landscape in an enterprise may also be an important deciding factor at arriving at the new infrastructure.

Solution analysis and design phase

Based on the inputs of "Requirements capture" on page 87, create an architectural overview diagram. An architecture overview diagram represents the governing ideas and candidate building blocks of an IT system and enterprise architecture. It has three levels:

- Enterprise view: This is similar to the SOA Foundation logical view in Figure 2-14 on page 55.

- Services view: This is similar to the SOA Foundation solution stack in Figure 2-13 on page 49 that explains how different layers of the solution relate to each other.
- IT systems view: This shows the physical building block of the solution, as shown in Figure 3-1 on page 77.

The diagram provides an overview of the main conceptual elements of the enterprise and their relationships, including candidates subsystems, their connections, and datasources. It is used as the starting building block for further identification of services and other application and technical components.

From the architecture overview diagram, you can build the Component Model, Component Interaction Diagram, and Operational Model.

- The Component Model (CM) describes the entire hierarchy of components, their responsibilities, relationships, and collaborations to deliver the required functionality. It is used as a primary concept for modular design and can be used to represent software, technological systems, or hardware components. It also captures the static relationship between components. Furthermore, components can be composed to form higher level components or into finer ones. For example, an ESB component can be decomposed into a business services directory, service metadata registry, message server, message router, and message broker components.
- The Component Interaction Diagram shows us how the components interact to realize the business process. Use Cases are mapped to the component model. The focus of this model is on the dynamic relationship between components, which is vital in understanding the application topology and its changes.
- The Operational Model (OM) focuses on describing the operation of the Information Technology system derived primarily from the non-functional requirements.

Figure 3-5 shows a sample component model for a stock trading company.

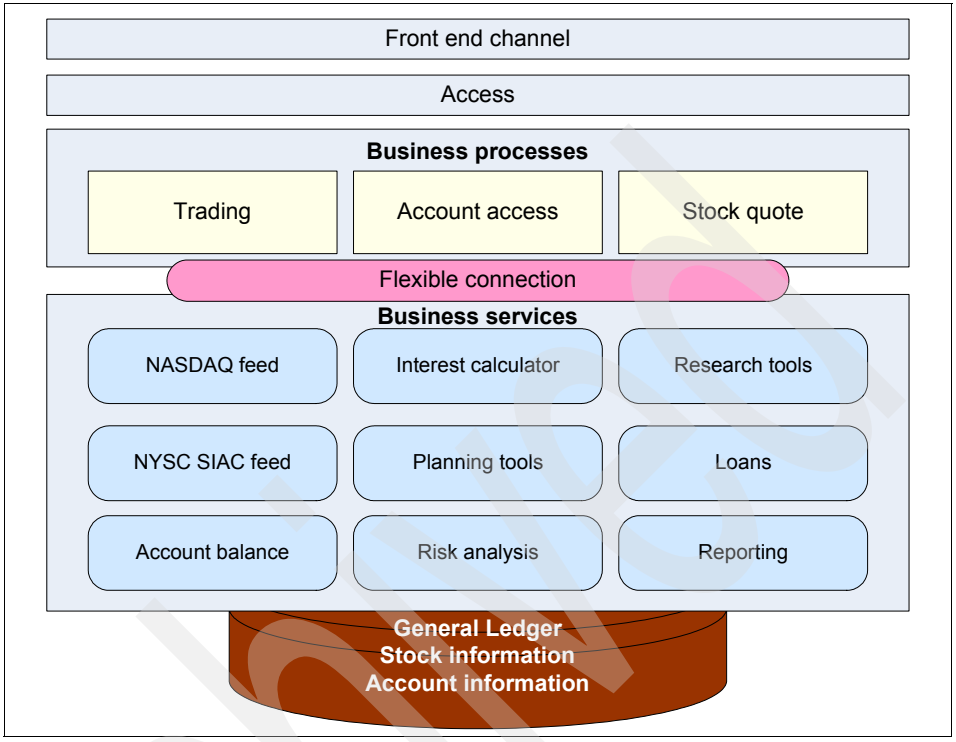


Figure 3-5 Sample component model

Map components to the SOA reference architecture

The components in the component model are mapped to the SOA reference architecture. This requires a thorough understanding of the role of the component along with other non-functional specifications. Figure 3-6 shows an example of this mapping.

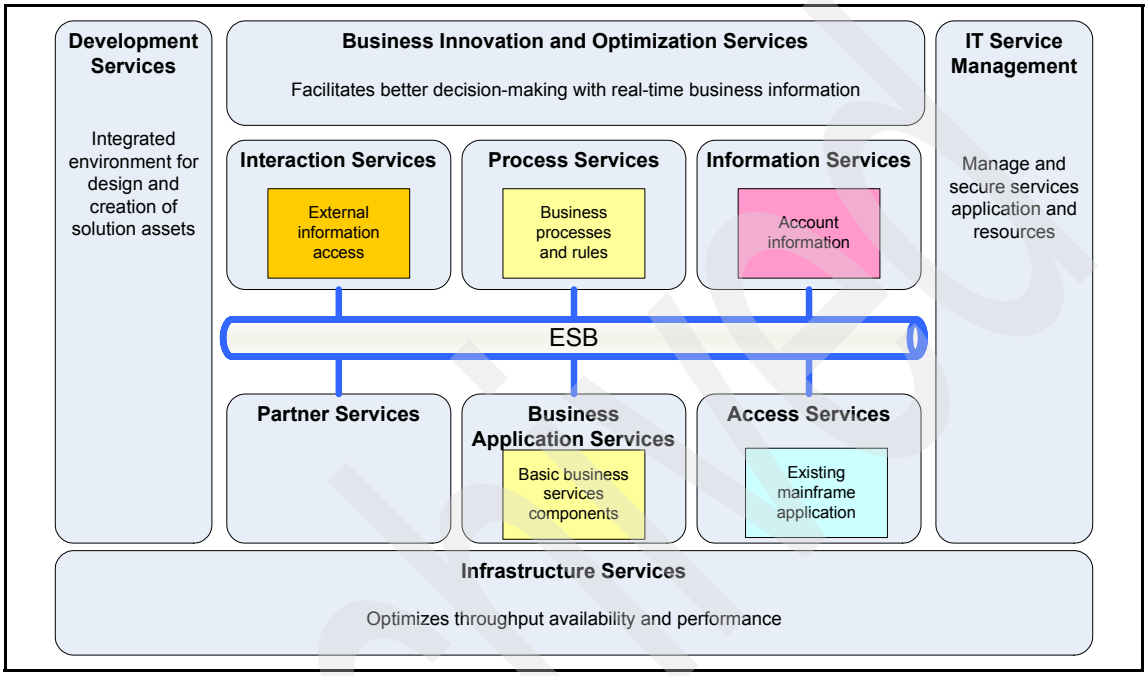


Figure 3-6 Mapping Component Model to SOA reference architecture

Document architecture decisions for integration

Based on the inputs from the component diagram, define and evaluate alternative choices for the communication mechanism among these components. Determine the best possible integration choice, taking into consideration the non-functional requirements, the location of the component, and its dependencies. Typical questions that would be asked at this stage are:

- I am integrating with Integration with .NET. Do I use messaging or a call interface?
- Do I use a JCA adapter when integrating with mainframe systems?
- Should ESB be the single integration hub or a federated integration?

Document the architectural decision of this stage, as shown in Figure 3-7 on page 91.

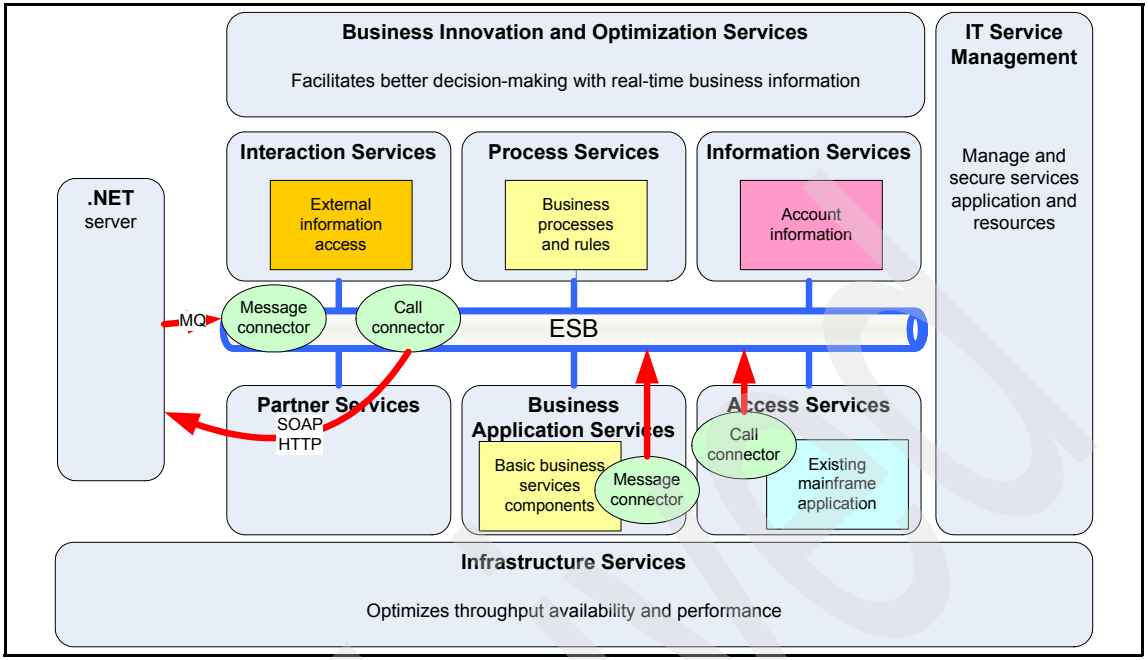


Figure 3-7 Architecture decisions for integration

Software and middleware product mapping

At this stage, evaluate the various middleware options available for the implementation of the components. Ensure that the product chosen at this stage fulfills all the requirements (functional and non-functional) of the application infrastructure. Typical questions that would be asked at this stage are:

- Which ESB platform do I choose?
- Which workflow engine do I use?
- What is the container for my Java applications?
- Which is the datastore that I should use?

As can be seen from these questions, this phase requires understanding the product functionality as well. There are many best practices and how-to guides, such as the ESB decision guide, which are available at IBM developerWorks, found at:

`http://www.ibm.com/developerworks/web/library/`

Figure 3-8 shows the product mapping phase.

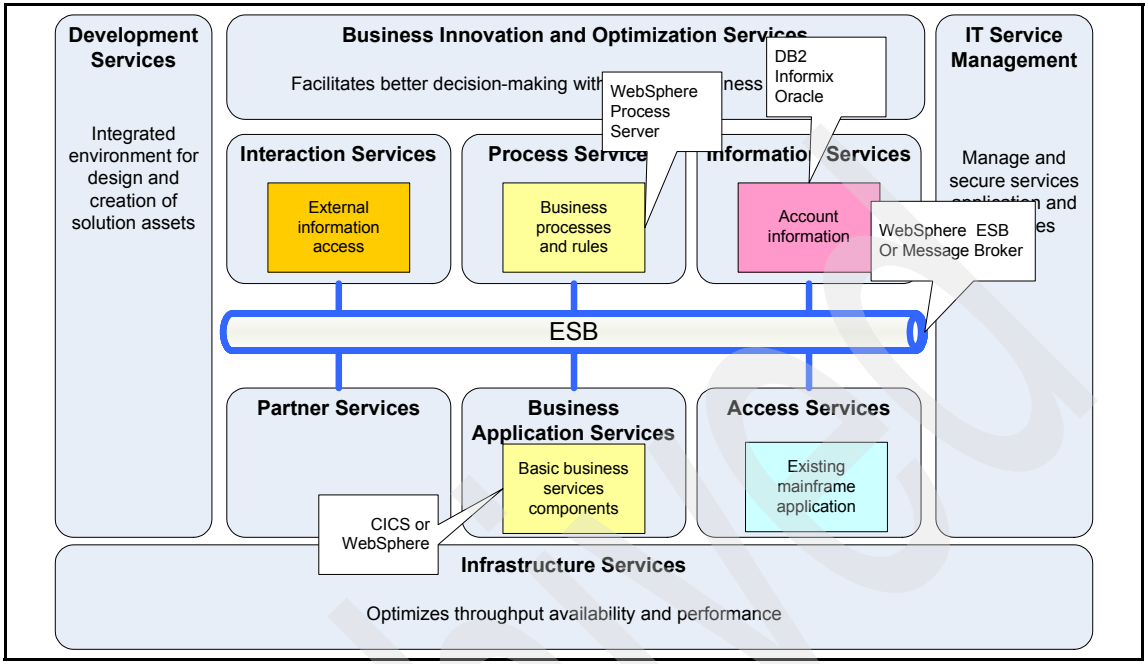


Figure 3-8 Product mapping phase

Define the Logical Operational Model

The next step is to define the logical operational model. This model identifies discrete layers of the enterprise and the positioning of the components within each of the layers, as shown in Figure 3-9 on page 93.

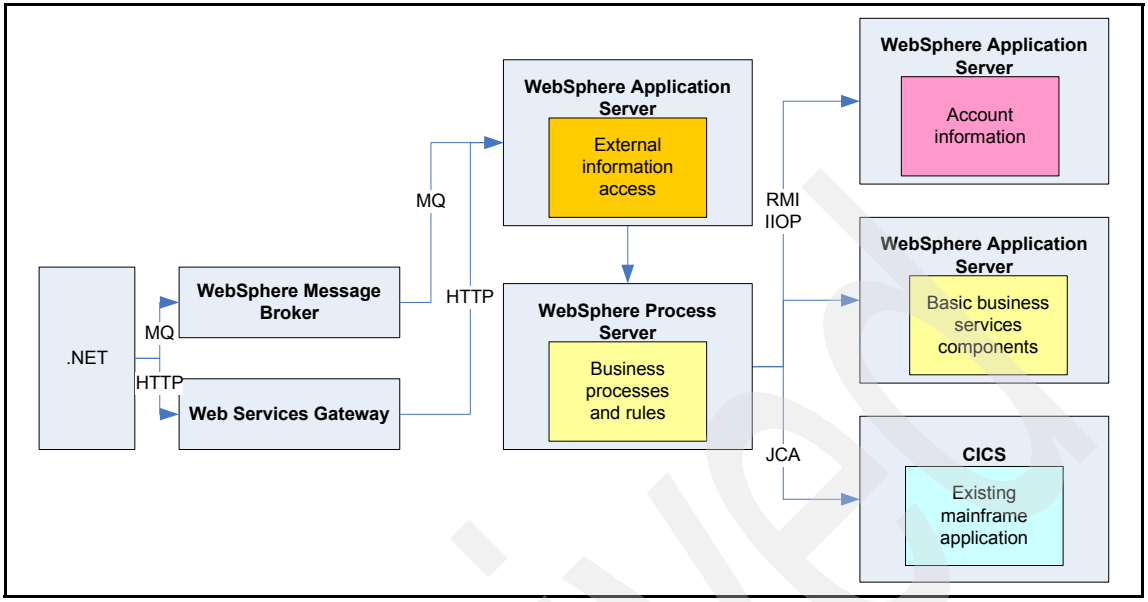


Figure 3-9 Define the logical operational model

Identify the deployment pattern for each product

For each of the identified products, determine the deployment pattern to be used. This can be done by analyzing the application properties along with the various non-functional requirements that the SOA application has to fulfill. For example, a business process that runs on WebSphere Process Server and requires extensive use of the Common Event Infrastructure with high throughput would require a high performance server.

Evaluate physical platforms

At this stage, evaluate the various physical platforms available. Define a set of scenarios to implement the physical operational model against the various physical platforms. A physical platform should already be considered along with the software platform selection. It also plays a part in the non-functional requirement of the product. Some software selection enforces the platform, such as .NET runs on Windows server and CICS runs on System z. However, other software selections are available for multiple platforms.

Platform selection should typically look at the reliability, availability, and scalability requirements. It also requires a good application sizing and modeling to be available. For the SOA case, a platform requirement includes the flexibility to scale and adapt to changes.

Figure 3-10 shows a sample platform selection for our logical operating model. It evaluates some combination of System p and System z platform implementation scenarios.

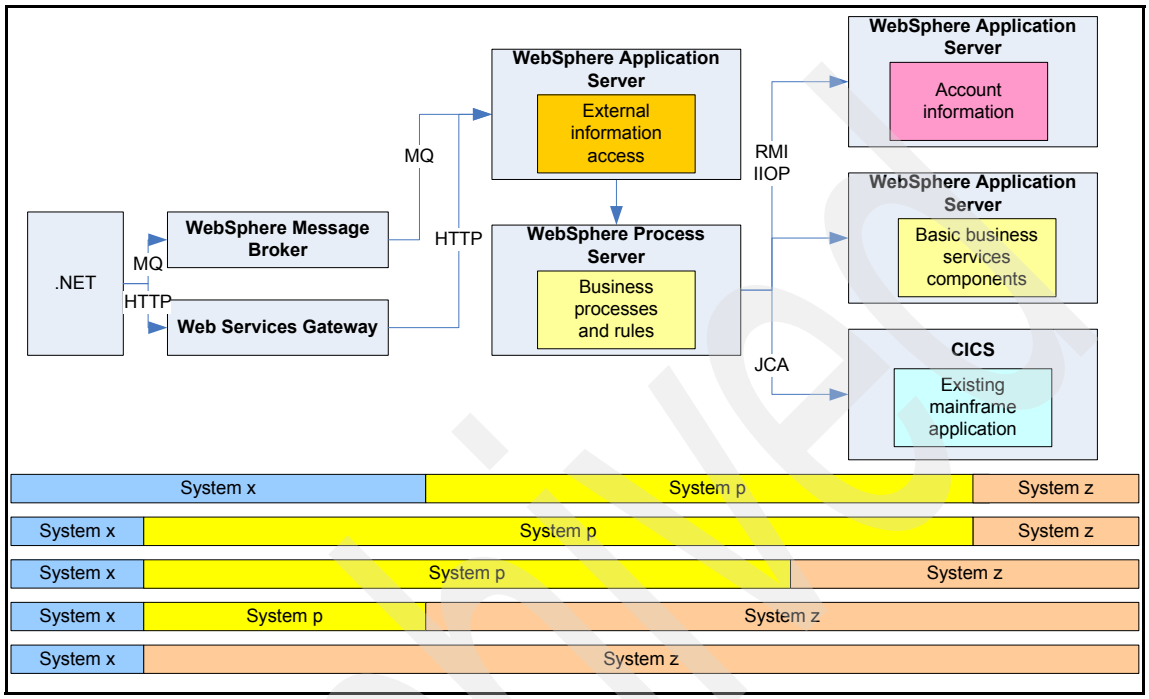


Figure 3-10 Evaluate physical platforms

Capacity planning

During this phase, replacement servers or partitions are sized. Factors such as current utilization, desired target utilization, growth factor, and spare pools were used to size each component. You may use statistical estimates and modeling tools to achieve a good result. You may also consider some sizing tools from the vendors to assist in this task.

Resource planning

In this phase, standards and guidelines are built for resource allocation based on the existing practices, type of component (Web, application, or database), applications to be run, bandwidth requirements, and best practices. The resources allocated to each server or partition are either dedicated or shared depending on these guidelines. Sizing information from capacity planning is used in this phase. Processor, memory, boot disks, and adapter requirements finalized in this phase are used in server configuration and ordering.

Resource management

Based on the resource planning, adapters, disks, and HBAs are assigned to LPARs. Multiple adapters of the same type required for a LPAR are assigned from multiple RIO units to eliminate RIO draw as a single point of failure.

3.4.2 Infrastructural anti-patterns

This section lists the key scenarios that, if not addressed, can become hindrances to architecting the right SOA infrastructure:

- Designing the application without looking at the infrastructure
- Leaving the infrastructure *as-is*
- Having the application and infrastructure groups not talk to one another
- Over-engineering the IT infrastructure
- Not updating any of the processes that are involved with the new applications
- Not focusing on the *basics* of good IT infrastructure and application development

3.5 System p and the infrastructure layer

Addressing issues as early as possible is important. Trying to fix infrastructural issues on a higher layer only adds complexity and mixes layer responsibilities. This also loses the clear structure of design and architecture; the model domains blur and get less structured.

As some of the business issues we identified through the process refer directly to infrastructural issues, the following System p benefits can address these issues directly on the lowest layer possible.

System p provides each of the partitions with the flexibility to extend or scale its resources based on the demand the business requires. Changes can be provided using partition settings or upgrade codes. In this section, we discuss some possible issues or demands the business or IT might have towards the application environment, such as:

- 3.5.1, “Abstract data access” on page 96
- 3.5.2, “Interacting with external resources” on page 96
- 3.5.3, “Provide new access points” on page 96
- 3.5.4, “Dynamic composition of business environments” on page 96

3.5.1 Abstract data access

As the source of data grows with new providers, an abstraction of the data access layer may be required. This abstraction can be implemented as a new service. The service's demands can be encapsulated in its own WPAR to compartmentalize its administration and impact on other running applications.

The additional load to the abstraction can be provided by moving the WPAR to another larger system partition as the need arises. Abstract data access can be separated from the overall application and even provided by its own operating environment. The abstract data access also allows additional providers to be integrated.

3.5.2 Interacting with external resources

Utilizing a virtual network adapter enhances SOA connectivity. Using virtual network adapters within a WPAR eases administration, as there is only one system to be configured.

A virtual network adapter allows easy network reconfiguration. If you need further separation of network access for administration or security purposes, the virtual adapter can be configured directly to the running system. Communication within the partitions is extremely reliable, with a very high throughput by using the internal communication mechanism in System p.

3.5.3 Provide new access points

It get easier to deploy new settings to add value and entry points to the whole SOA framework. The horizontal scale of a particular system can be increased on demand with virtualization facilities of the partitions. Additional systems can be integrated quickly by creating a new access point, such as mobile access for external users.

3.5.4 Dynamic composition of business environments

The SOA environment is dynamic. The infrastructure supporting SOA should be able to keep up with the business changes that can be accommodated by SOA. If the infrastructure is unable to keep up with the application changes, the SOA benefit cannot be achieved. System p virtualization allows deploying new servers and new services easily, either through a compartmentalized WPAR or by performing a real system partition. The SOA infrastructure changes can be incorporated to adopt to changes in application flows and demand.

3.6 Conclusion

SOA projects must deal with transformation of the way companies look at IT and look at their applications. Today's SOA infrastructure must provide the same services for all applications. We are not providing infrastructure capabilities per application, but providing them to the enterprise so that all services that support the business can leverage that infrastructure. A well designed SOA infrastructure is a mix of current and SOA infrastructure technologies. Understanding how these services work is the key to architecting and building successful SOA infrastructures.

If the infrastructure does not keep pace with the SOA applications, what is known as a risk app can emerge, leading to shortcomings in the infrastructure's ability to support, operate, and manage the application. The goals of agility and flexibility will actually be lost and possibly result in the infrastructure design being less able to adapt due to complexity. So it is not acceptable to ignore the infrastructure and only address the application aspects of SOA. You do need to make sure the infrastructure keeps pace with the changing paradigm of SOA.

4

IBM System p features

This chapter will introduce the synergy between Power Systems and SOA. The discussion mainly concerns the AIX and Linux operating systems. We discuss System p features and highlight which ones are especially relevant for SOA. Then we describe the benefit to your SOA infrastructure. The chapter is divided into the following sections:

- 4.1, “System p and SOA synergy” on page 100
- 4.2, “System p feature overview” on page 103
- 4.3, “Operating system features overview” on page 125
- 4.4, “Realization of System p benefits to SOA” on page 147

4.1 System p and SOA synergy

This section discusses how a specific system platform may influence an architectural style for SOA and why this specific platform is ideal for SOA and its operation. SOA, as a mission critical application, requires reliability from the hardware platform. It also requires hardware and operating system facilities that can lead to better security of the application, such as a high entropy random generator.

We show how System p is able to create an architectural synergy between the IT infrastructure and the abstract services level of SOA through its virtualization capabilities.

4.1.1 System p architecture

This section introduces the System p architecture. We mainly focus on the system structure and its virtualization capabilities that are beneficial to SOA.

Figure 4-1 shows a simplified System p architecture. In the center, POWER Hypervisor™ provides each virtual partition with the resources and connectivity to operate. Adding, changing, and removing resources is easy, as the system provides dynamic scalability and reconfiguration.

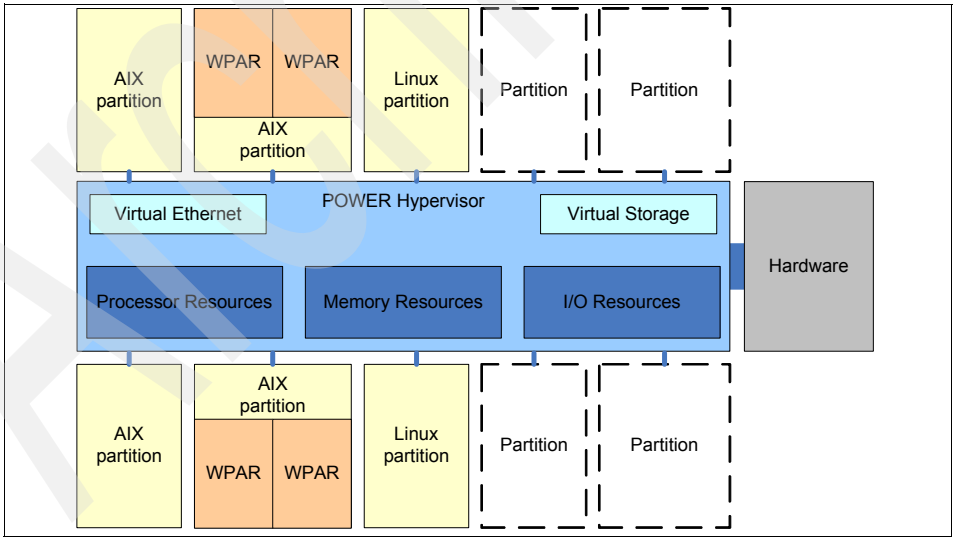


Figure 4-1 System p architecture

The architecture is similar to the bus concept, with the POWER Hypervisor providing an abstraction of the hardware towards the partitions and therefore the software running on these partitions. Specific abstractions like storage and network are provided by virtual network and storage interfaces that can be arranged and configured independent from the actual hardware.

Refer to *Advanced POWER Virtualization on IBM eServer p5 Servers*, SG24-5768 can show further system and configuration details.

4.1.2 SOA and System p architectural synergy

Recalling the architectural view of SOA in Figure 2-14 on page 55, we notice a similarity between the System p architecture and SOA architecture. Referring to both architectures, the bus concept can be seen as a common layer in both, even though a clear separation of service consumers and providers exists in the System p architecture. The hardware of a service provider is abstracted by the POWER Hypervisor towards the service consumers: the partitions.

Figure 4-2 illustrates this synergy between a hardware platform and an architectural style applied primarily to software.

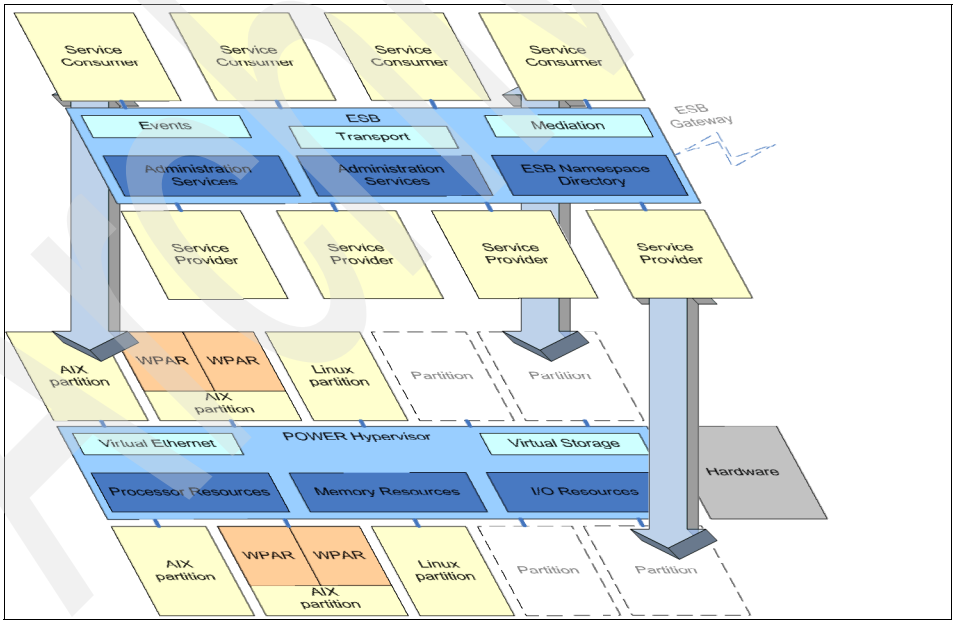


Figure 4-2 Synergy of IT infrastructure and ESB architecture

Even though the SOA middleware layer supported by an ESB is not able to run on any hardware directly, the abstract layer of the POWER Hypervisor allows for a perfect composition of the partitions towards SOA components decoupled by a middleware like ESB. Figure 4-2 on page 101 illustrates a mapping of service provider or consumer to a partition. In real scenarios, many services may reside on a partition but the architectural synergy allows for an ideal composition of them.

This architectural synergy enables a clear structure between the logical view of services and involved hardware. Seeing other aspects of the operation, such as performance, it is clear why this can be important. Enabling independent tuning based on a logical view of services, for example, can provide a huge benefit towards the whole solution.

The support POWER Hypervisor gives to the SOA seems only domain specific. Having a server consolidation across an enterprise, the communication of the whole SOA gets more manageable and structured, as Figure 4-3 shows.

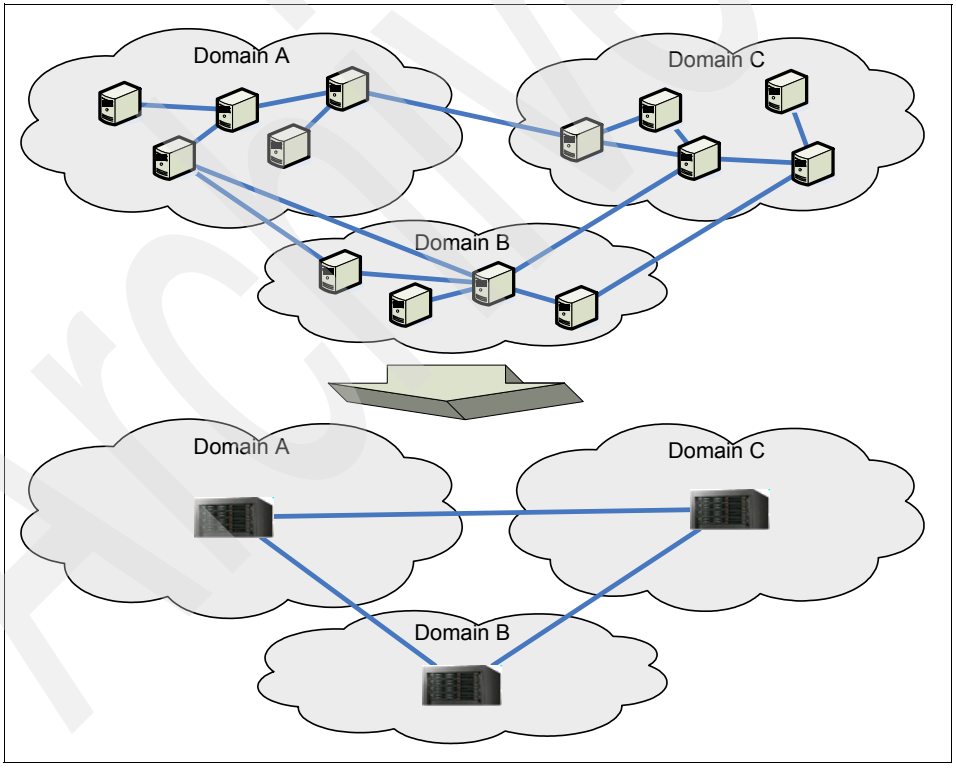


Figure 4-3 System p infrastructure consolidation

Based on the architectural synergy and infrastructure consolidation, we show specific System p aspects and how they relate to SOA in the next section. This includes some more hardware specific aspects and features of System p and the operating system hosted by it.

4.2 System p feature overview

This section shows some System p features, as shown in Figure 4-4. These features have a *value add* towards a flexible and business aligned IT infrastructure provided by System p. The significance of the features for the individual case may vary depending on the requirements and business needs.

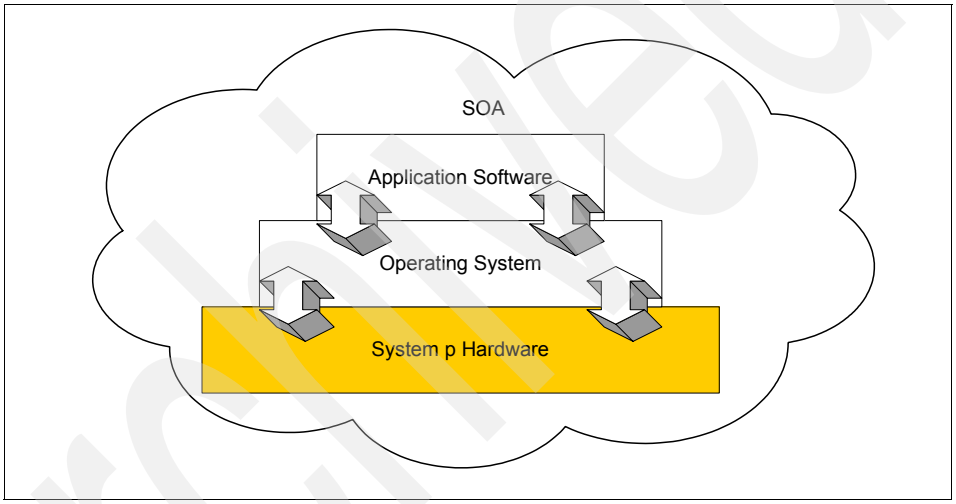


Figure 4-4 Feature overview: System p hardware

Figure 4-4 illustrates the hardware as a supporting layer for the operating system with bi-directional communication. The communication includes direct usage of hardware and feedback from the operating system towards the hardware.

The discussion includes:

- 4.2.1, “Virtualization on System p” on page 104
- 4.2.3, “System p availability features” on page 116
- 4.2.4, “System p management features” on page 119

4.2.1 Virtualization on System p

Virtualization is usually associated with software virtualization, where a software emulates the whole environment and is itself run by an environment. Such types of virtualization can be considered a mighty tool. But regardless of their power and benefits, they usually lack in performance, scalability, and relocation aspects.

System p includes reinforced virtualization concepts in its core concepts and architecture. This allows System p to provide a high level of system performance and the promotion of advanced virtualization concepts towards upper layers. Additionally, System p provides high levels of availability, service quality, and extensive autonomic features, referring to the integration of artificial intelligence technologies to tasks.

PowerVM on System p offers industry leading virtualization capabilities for AIX and Linux. PowerVM is the new umbrella name for Advanced Power Virtualization. The PowerVM editions are optional hardware features available on IBM System p servers based on POWER5™ or POWER6™ processors. PowerVM comes in three editions:

- PowerVM Express

  The PowerVM Express edition is available on the POWER6 technology-based System p550 and System p520 Express servers and includes these features:

  - Up to three partitions per server
  - Shared dedicated capacity
  - Integrated Virtualization Manager
  - Virtual I/O Server

- PowerVM Standard

  The PowerVM Standard Edition is a combination of hardware enablement and software that includes the following components:

  - Micro-partitioning
  - Shared dedicated capacity
  - Virtual I/O Server
  - Shared Ethernet adapter
  - Virtual SCSI
  - Integrated Virtualization Manager

  PowerVM Standard Edition on POWER6 systems adds the following:

  - Higher maximum number of LPARs
  - Multiple shared processor pools

Virtual Ethernet and dedicated-processor LPARs are available without this feature for servers attached to an HMC or managed using the IVM. These hardware feature can be activated initially or upgraded by a firmware key.

- PowerVM Enterprise

  PowerVM Enterprise edition is only available on POWER6-based systems. The PowerVM Enterprise edition includes all of the standard edition features, with the addition of Live Partition and Application Mobility. Live Partition Mobility allows you to migrate running AIX and Linux partitions and their hosted applications from one physical server to another without disrupting the infrastructure services. The migration operation maintains complete system transactional integrity. The migration transfers the entire system environment, including processor state, memory, attached virtual devices, and connected users. For more information about the Live Partition Mobility of the PowerVM Enterprise Edition, refer to *PowerVM Live Partition Mobility on IBM System p*, SG24-7460.

To find the current activation codes for PowerVM for a specific server, use the IBM Web site:

`http://www-912.ibm.com/pod/pod`

All IBM PowerVM editions support PowerVM Lx86.

IBM PowerVM Lx86

PowerVM Lx86 brings additional benefits to customers and application providers who want the reliability and flexibility of System p in their Linux environment yet use applications that have not yet been ported to the platform. PowerVM Lx86 allows most Linux x86 binaries to run unchanged on POWER systems using translation software that translates x86 instructions dynamically to POWER instructions and transforms x86 Linux system calls into calls to the POWER Linux kernel. The IBM PowerVM Lx86 includes features that enable the consolidation of AIX 5L™ and Linux on POWER applications, plus most x86 Linux applications, on the same server.

By adding x86 applications to System p Linux solutions, businesses and independent software vendors (ISVs) can expand application use in consolidation scenarios. In these scenarios, businesses can maximize the use of their hardware investments and improve total cost of ownership by reducing time-consuming tasks that are involved in managing multiple servers. At the same time, they can enhance service quality and mission-critical application availability.

PowerVM Lx86 provides an environment in which the benefits of running on System p can be available to customers who need to run SOA applications currently available only on x86 platforms. No application changes are required for customers to get the following benefits:

- Outstanding IBM System p reliability, availability, and serviceability
- PowerVM features that can offer dramatic savings in hardware and systems management for multiple servers

Applications have enhanced System p options for growth scenarios, such as:

- Scale up: Increasing performance by adding more processors and memory
- Scale out: Increasing performance by adding additional complete systems
- Scale within: Consolidating multiple workloads onto a single server utilizing virtualization

PowerVM Lx86 is transparent to the x86 application. You can get more information about PowerVM Lx86 from *Getting Started with PowerVM Lx86*, REDP-4298.

Hypervisor

The POWER Hypervisor is the foundation for virtualization on a System p server. It enables the system to be divided into multiple partitions and ensures strong security and isolation between them. The POWER Hypervisor also enforces partition security and enables the Virtual I/O Server's virtual SCSI and virtual Ethernet features.

Live Partition Mobility

Live Partition Mobility, licensed through PowerVM Enterprise Edition, is a feature that allows you to move running AIX or Linux partitions from one physical POWER6 server to another without disruption to the running application. Some possible uses and advantages of Live Partition Mobility are:

- Moving partitions based upon workload demands to maintain an optimal level of service and cost
- Consolidation of underutilized partitions to enable underutilized servers to be taken offline, saving power and cooling costs
- Turning disruptive maintenance procedures into nondisruptive ones
- Meeting stringent service-level agreements through proactively moving running partitions and applications from one server to another
- To better balance workloads and resources

For more information about Live Partition Mobility, see *PowerVM Virtualization on IBM System p: Introduction and Configuration Fourth Edition*, SG24-7940.

Workload Partitions (WPAR)

AIX V6.1 introduces a new, software based, virtualization approach called Workload Partitions (WPARs). WPARs enable the creation of multiple virtual AIX V6.1 environments inside of a single AIX V6.1 instance. WPARs share a regulated portion of the processing and I/O resources of the global instance but are isolated from the processes and users in other WPARs or in the global instance. Each WPAR can have its own:

- Unique “root” administrator
- Network addresses
- File systems
- Security context (users and groups).

There are two types of Workload Partitions: System and Application.

- System WPARs look like independent AIX V6.1 instances. They have their own copies of many system services like init and mail, they can be logged into via telnet, and they have their own users and groups.
- Application WPAR is simply a wrapper around an application that makes it more manageable. Application WPARs run inside of the global instance and do not have their own administrator, file systems, or security context. All processes running inside of an Application WPAR can be grouped together for management, including resource controls. Because Application WPARs are not running their own copies of system processes like init, they have an even smaller resource footprint than System WPARs.

IBM Workload Partitions Manager™ for AIX (WPAR Manager) lets you manage WPARs across multiple systems. The WPAR Manager product is available separately; it is not part of AIX V6.1.

Workload Partitions can be moved from one system to another without restarting the application or causing significant disruption to the application user. This process is called Live Application Mobility, a feature of AIX V6.1 and the Workload Partitions Manager for AIX (WPAR Manager). During the relocation process, the WPAR Manager first creates a checkpoint of the Workload Partition, and then the memory and other WPAR configuration information is moved to the target system, and finally, the WPAR is resumed on the new system, right where it left off. Applications do not have to be restarted because the entire WPAR, including the application context, has been moved to the target system.

4.2.2 Logical partitioning

The logical partition within a System p server is built upon the POWER Hypervisor, ensuring an isolated and secure hardware-based virtual environment for the operating system to run on. A logical partition definition describes the amount of memory, processing capacity, and I/O adapter resources that are assigned to the operating system image.

It is possible to dynamically move system resources between partitions without rebooting. This is known as dynamic reconfiguration or dynamic logical partitioning (DLPAR). The following are resources that can be dynamically moved:

- Physical processors (for dedicated processor LPARs)
- Virtual processors or entitled capacity (for micro-partitions)
- Memory
- Physical I/O adapters
- Virtual I/O adapters

This section discusses the following topics:

- "Processor virtualization" on page 108
- "Memory virtualization" on page 112
- "I/O virtualization" on page 112

Processor virtualization

System p servers provide a number of virtualization features to implement the partitioning of processors. These diverse features allow the system a high degree of flexibility in configuring systems for a SOA. Figure 4-5 on page 109 shows the concepts of processor virtualization that we discuss in this section.

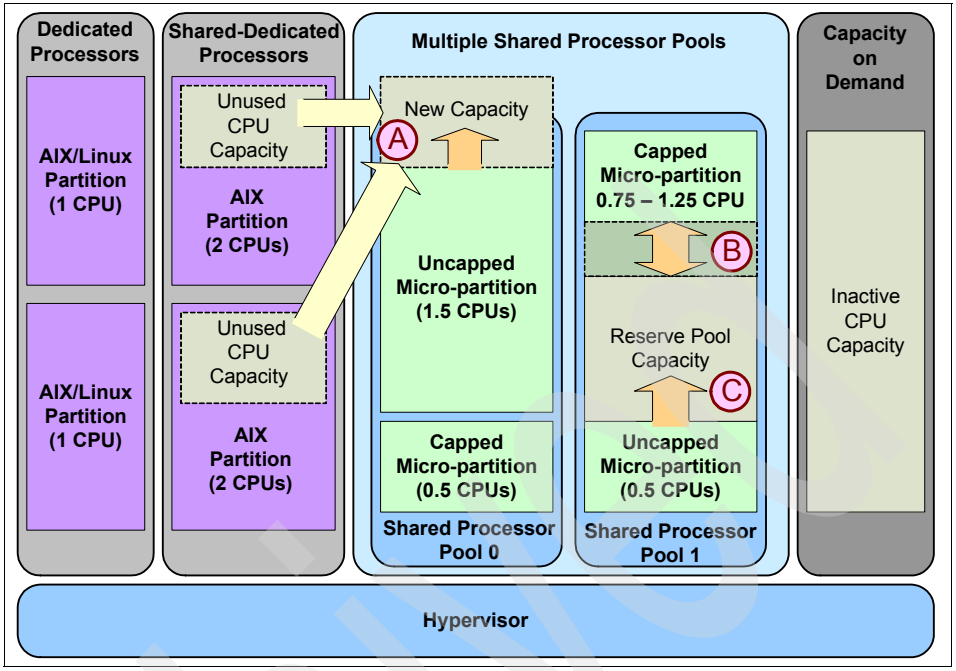


Figure 4-5 System p processor virtualization features

The following is a list of the processor virtualization features available on System p servers:

- Micro-partitioning

  Micro-partitioning allows processors to be divided into multiple fractions of a processor. This feature is a key factor in making the most of a server's processing capacity. When creating or changing a partition, a processor may be allocated in units of one-tenth (1/10th) of a processor. A partition may be allocated as little as one-tenth (1/10th) of a processor or up to as much as is available on the entire system. Through dynamic reconfiguration or dynamic logical partitioning, processors can be subdivided as finely as one one-hundredth of a processor.

- Shared dedicated capacity

  This new configuration option available on POWER6 systems enables the administrator to donate excess processor cycles from a dedicated processor partition to a Shared Processor Pool without affecting the workload running in the dedicated processor partition. Refer to Figure 4-5 to see how the unused capacity is donated to a shared processor pool.

As you can see from the figure, highlighted by notation A, this new capacity in the shared processor pool allows the pool to exceed its entitled capacity, thus allowing a system to effectively provide more capacity than is installed in the system.

- Virtual processors

  System p partitioning technology allows an allocation of partial physical processor to a partition. This partial physical processor is called a virtual processor.

- Shared-processor pooling

  Shared-processor partition capability was introduced with IBM POWER5-based servers. These shared-processor partitions, known as micro-partitions, are allocated processor resources from a single pool of physical POWER processors referred to as the physical shared-processor pool. The entitled capacity of a micro-partition may range from ten percent (10%) of a physical processor up to the entire capacity of the physical shared processor pool. The physical shared processor pool can have from one physical processor up to the total installed processor capacity of the system. Changes to the entitled capacity of a micro-partition can be as granular as one percent (1%) of a physical processor. Micro-partitions can be designated either capped or uncapped. These modes are defined as follows:

  Capped A capped micro-partition has a defined processor entitled capacity that it is guaranteed to receive. There are no circumstances under which a capped micro-partition will receive any more processor resource that its entitled capacity.

  Uncapped An uncapped micro-partition has a defined processor entitled capacity that it is guaranteed to receive. However, under some circumstances, it can receive additional processor resources. If the uncapped micro-partition is at full CPU load and there are unused processor cycles within the physical shared processor pool, then additional cycles are automatically allocated to the micro-partition on a weighted basis.

- Multiple Shared-Processor Pools

  POWER6-based systems introduce support for multiple shared-processor pools. This new capability allows a systems administrator to create a set of micro-partitions with the purpose of controlling the processor capacity that they can consume from the physical shared-processor pool. Each shared-processor pool has an associated Entitled Pool Capacity that is consumed by the set of micro-partitions in that Shared-Processor Pool.

The micro-partitions within a Shared-Processor Pool are guaranteed to receive their entitled capacity. In addition, unused processor cycles within their Shared-Processor Pool are harvested and are redistributed to eligible micro-partitions within the same Shared-Processor Pool. The source of additional processor cycles can be the Reserved Pool Capacity (see notation “C” in Figure 4-5 on page 109), which is processor capacity specifically reserved for a Shared-Processor Pool but not assigned to any of the micro-partitions in the Shared-Processor Pool.

When the set of micro-partitions in a Shared-Processor Pool are heavily loaded, they can consume additional processor capacity (assuming they are uncapped) from outside their Shared-Processor Pool up to a defined maximum, Maximum Pool Capacity. Processor capacity distributed in this way have been ceded by underutilized sets of micro-partitions in their Shared-Processor Pools.

- Capacity on Demand (CoD)

  Capacity on Demand gives you a variety of ways for reserve CPU and memory to be purchased as needed. The following are types of on demand capacity:

  Capacity Upgrade on Demand

  Inactive capacity is designed into the original purchase, with little, if any, up-front charges or pricing premiums for this additional capacity. You can pay for these upgrades when they are activated

  On/Off Capacity on Demand

  On/Off Capacity on Demand helps businesses cope with both predictable and unpredictable surges in transaction volume with the aid of temporary increases in processor capacity. You can activate and deactivate quickly and efficiently as the demands on your organization dictate: pay for what you need, when you need it, and the server will keep track of your usage. On/Off Capacity on Demand provides a highly cost-effective strategy for handling seasonal or period-end fluctuations in activity and can enable you to deploy pilot applications without investing in new hardware.

Trial Capacity on Demand

Trial Capacity on Demand allows businesses to have up to 30 contiguous days of additional processing power, employing up to the maximum number of processors and memory installed on the server. You can activate Trial Capacity on Demand to test new applications, determine the effect of added computing resources on your existing workload, or support your business while in the process of purchasing permanent capacity upgrades.

Capacity BackUp Capacity BackUp adds temporary processing capacity to a backup server in the event of an unforeseen loss of production server capability due to forces beyond your control. With Capacity BackUp, you can divert entire workloads to backup servers for up to 90 days. There is an upfront fee and an emergency-use fee, which include testing; software fees are not affected.

Memory virtualization

Memory is allocated to the logical partitions on the System p platform from the overall memory that is installed in the system. The amount of memory is determined during capacity planning.

For each partition on the managed system, the Hypervisor sets aside a small amount of memory to manage the memory addressing for the partition. This amount is based on the maximum memory definition for the partition and is 1/64th of the value entered as the maximum value in the Hypervisor firmware memory.

I/O virtualization

The partitioning of I/O adapters into virtual devices that client partitions can then use is enabled by the Virtual I/O Server. The Virtual I/O Server is a stripped down appliance AIX image that provides for the virtualization of I/O adapters. From within the Virtual I/O Server physical resources are shared and served up as virtualized resources among a group of client AIX or Linux logical partitions. The Virtual I/O Server virtualizes storage and network adapters through the Virtual Ethernet, Virtual SCSI, and Virtual adapter functions.

Using the VIOS facilitates the following functions:

- Sharing of physical resources between logical partitions on the system
- Creating more logical partitions without requiring additional physical I/O resources

- Creating more logical partitions than there are I/O slots or physical devices available with the ability for partitions to have dedicated I/O, virtual I/O, or both
- Maximizing use of physical resources on the system
- Helping to reduce the Storage Area Network (SAN) infrastructure

Virtual Ethernet

Virtual Ethernet provides high speed network communication (from 1 to 3 Gbps) between LPARS. Virtual Ethernet supports Virtual LAN (VLAN) capability. VLAN allows secure communication between logical partitions. Bridging network traffic from the virtual Ethernet networks out to physical Ethernet networks is performed through the creation of Shared Ethernet Adapters. Figure 4-6 shows a simple Virtual Ethernet configuration.

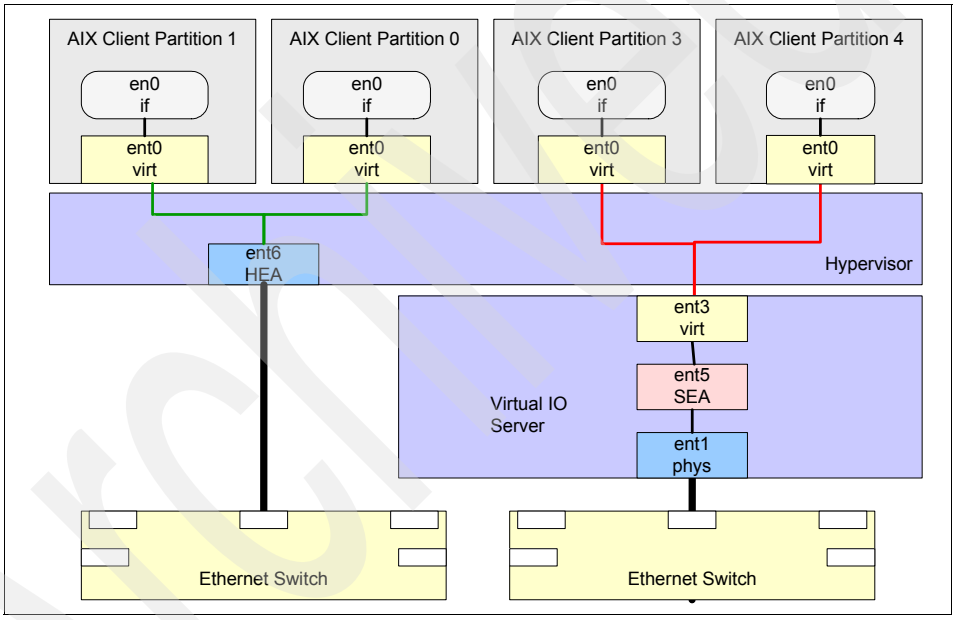


Figure 4-6 Ethernet configuration showing both HEA and SEA options

- Host Ethernet Adapter

  The Host Ethernet Adapter (sometimes referred to as the Integrated Virtual Ethernet) is a new resource on POWER6 systems that allows partitions to have a physical Ethernet port without utilizing the Virtual I/O server. In Figure 4-6, ent6 is the Host Ethernet Adapter that allows AIX Partition 1 and AIX Partition 2 to share a physical Ethernet port to an external LAN. Note how this sharing does not require a Virtual I/O server.

- Shared Ethernet Adapter

  The Shared Ethernet Adapter (SEA) is a physical network connection that routes network traffic from a Virtual Ethernet to an external network and thus removes the need for a physical network adapter attached to an LPAR. In Figure 4-6 on page 113, the physical Ethernet port ent1 is in the Virtual I/O Server. Through ent5, the Shared Ethernet Adapter, virtual Ethernet port ent3 is made available for AIX Client partition 3 and AIX Client Partition 4 to access an external network.

Virtual SCSI

Virtual SCSI facilitates the consolidation of LAN and disk I/O resources and minimizes the number of physical adapters that are required. Virtual SCSI is served through the Virtual I/O Server, which provides storage backing devices. The combination of virtual SCSI and the Virtual I/O Server capabilities allows you to share storage adapter bandwidth and (optionally) to sub-divide single large disks into smaller segments. The adapters and disks can then be shared across multiple partitions. See Figure 4-7 on page 115 for an architectural diagram of virtual SCSI.

The following are supported as Virtual SCSI backing devices:

- Direct attached entire disks from the Virtual I/O Server
- SAN disks attached to the Virtual I/O Server
- Logical volumes defined on either of the previous disks
- File backed storage, with the files residing on either of the first two disks
- Optical storage devices

Supported storage devices are listed at the following site:

http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/datasheet.html

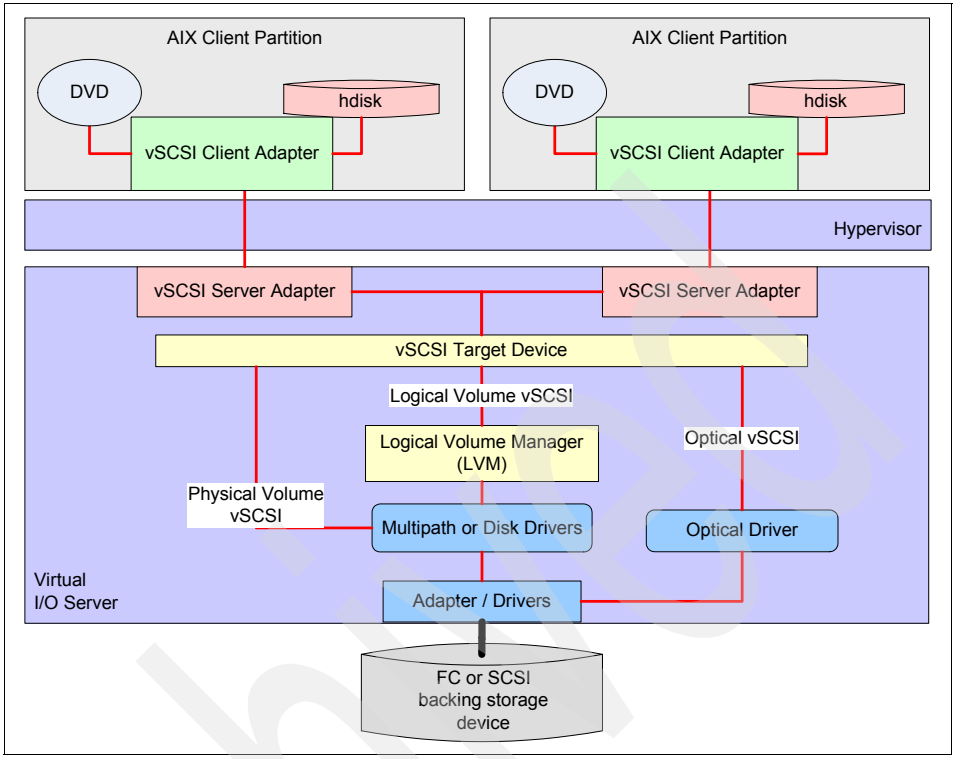


Figure 4-7 Virtual SCSI architecture

Logical partitioning and Linux

Linux is supported for running in one or more static logical partitions (LPARs) on all System p servers that support logical partitioning. The AIX 5L and Linux operating systems can run concurrently in separate partitions on an LPAR-enabled system in any combination. This capability enables a client to consolidate workloads from several separate servers onto a single system and increase the system utilization. Since the partitioning is controlled by the Hypervisor firmware and the Hardware Management Console (HMC) or the Integrated Virtualization Manager, the AIX 5L operating system is never required to run Linux.

Dynamic LPAR is not supported by Linux 2.4 kernel-based distributions or on POWER4™ processor-based systems. However, Linux partitions can be created on systems enabled for dynamic LPAR. The Linux partition appears grayed out on the HMC and cannot be dynamically changed on POWER4 systems. On POWER4 systems, to reconfigure Linux in an LPAR environment, it must first be stopped, the partition reconfigured, and Linux restarted.

Important: After installing Linux (Red Hat or SUSE®), you must download and install the IBM service and productivity tools packages to support dynamic LPAR functions. These packages include the Resource Monitoring and Control (RMC) daemon, which communicates with the HMC. You can download this utility at:

https://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html

4.2.3 System p availability features

This section explains availability features on IBM System p. It basically discusses the capabilities for reliability, availability, and serviceability (RAS).

The RAS design philosophy employs an architecture-based design strategy that seeks to avoid application outages. However, if a fault does occur, the system identifies, analyzes, and isolates the failing component through dynamic "self-healing" or through standard service practices as quickly as possible with little or no system interruption.

The IBM System p RAS approach in a nutshell is:

1. Avoid problems where possible.
2. Attempt to recover or retry operation failures.
3. Diagnose problems and reconfigure the system as needed.
4. Automatically initiate a repair and call for service.

Reliability

Reliability refers to the ability of a system or component to perform its required function, under stated conditions, for a specified period of time. Reliability is defined as the probability of how frequently a defect or fault in the system manifests itself.

Availability

Availability is defined as keeping a total system and its applications available to the users while maintaining a balance among performance, cost, and reliability. Availability deals with how infrequently the functionality of a system or subsystem is impacted by a defect or fault. System p enables a higher level of availability by enabling physical resources to be removed, upgraded, or changed without affecting users.

Components of the System p availability strategy are:

- First Failure Data Capture

  FFDC is a technique that ensures that when a fault is detected in a system, the root cause of the fault will be captured without the need to recreate the problem or run tracing or diagnostics programs. FFDC is a critical component to System p self-diagnosing and self-healing capabilities. The server uses thousands of diagnostic probes that can trap hardware errors at run time to further enable fault detection and isolation. These system checks support a strategy of predictive failure analysis by providing the ability to track intermittent correctable errors, thereby allowing the system to take components offline before they cause a crash.

- Automatic detection and deallocation of failing components

  Runtime correctable and recoverable errors are monitored to determine if there is a pattern of errors or a trend towards uncorrectability. Should these components reach a predefined error limit, the system will initiate an action to deconfigure the faulty hardware, helping avoid a potential system outage and enhancing system availability.

  Persistent component deallocation

  This is when a component that is identified for deallocation or deconfiguration on a POWER6 server is marked for persistent deallocation. Component removal occurs at run time or at boot time (during initial program load).

  Dynamic processor deallocation

  Enables automatic deconfiguration of an error-prone processor core before it causes an unscheduled server outage. The system is able to keep running due to the following:

  – If possible, an unlicensed Capacity on Demand (CoD) CPU will, by default, be automatically used for Dynamic Processor Sparing.

  – If no CoD processor core is available, an un-allocated core is used.

  – If no spare processor core is available, an attempt to locate a sum total of 1.00 spare processor capacity from an enabled shared processor pool is performed.

- If there is still not enough capacity, the system will determine how many processor core capacity units each partition needs to relinquish to “build” a processor from a shared pool, and then notify the operating system that processor units or virtual processors need to be varied offline.

- POWER6 processor availability enhancements

  POWER6 processor-based systems contain a suite of mainframe-class recovery features that significantly reduce scenarios that would result in failure.

  - Processor instruction retry
  - Alternate processor recovery
  - Partition availability priority
  - Processor contained checkstop

- Memory protection enhancements

  A variety of different protection schemes are used in IBM POWER6 processor based systems to avoid uncorrectable errors in memory.

  - Memory Persistent Deallocation occurs if defective memory is discovered at boot time. If a memory fault is detected at boot time, the affected memory will automatically be switched off and will not be used on subsequent reboots.
  - Hardware Scrubbing, which is a method of dealing with transient, or soft, errors.
  - Error Correcting Code (ECC) allows a system to detect up to two errors in a memory word and the ability to correct one of these errors.
  - Chipkill™ is IBM’s proprietary enhancement of ECC that enables a system to sustain the failure of an entire DRAM.
  - Redundant Bit Steering is IBM’s means of avoiding situations where multiple single-bit errors align to create a multi-bit error. This both reduces exposure to multi-bit errors as well as helps to defer maintenance.
  - Memory Page Deallocation POWER6 processor-based systems are able to recover from coincident single cell errors in separate memory chips using a memory page deallocation scheme for partitions running AIX and for memory pages owned by the POWER Hypervisor.

- PCI error recovery

  PCI error recovery uses a combination of system firmware and Extended Error Handling device drivers that allow recovery from intermittent PCI and PCIe bus errors. This approach initiates system recovery for a permanent PCI bus error rather than failing immediately; the faulty device is restarted, preventing downtime.

- Uncorrectable error handling

  System p servers contain additional recovery features to attempt to recover from uncorrectable errors. When an uncorrectable error does occur, sometimes the error is transient in nature and occurs in data that can be recovered. In cases where the data cannot be recovered from another source, Special Uncorrectable Error handling is used to determine whether the corruption is truly a threat to the system or not. If the data is never actually used, then the error condition can safely be voided and the system will continue to operate normally.

- Concurrent firmware update

  Concurrent firmware maintenance allow you to apply firmware updates on a running system without shutting down and restarting logical partitions, servers, or stopping applications.

For more information about POWER6 availability features, refer to the following Web site:

http://www-03.ibm.com/systems/p/hardware/whitepapers/power6_availability.html

4.2.4 System p management features

This section discusses some System p management features. The management of capabilities is a key success factor towards an effective IT infrastructure. System p support for management tasks is a major feature to ease enterprise IT management in a target oriented and sophisticated manner. In this section, we provide an overview of System p features that simplify current IT infrastructures and their associated management.

Hardware Management Console

The HMC provides a centralized point of hardware control in a System p environment. A single HMC can manage multiple System p servers, and two HMCs can manage the same set of servers in a dual-active configuration that is designed for high availability.

The major HMC functions include:

- Management of LPARs, including power on, power off, and console
- Creation of logical partitioning with dedicated processors
- Monitoring of system status
- Management of IBM Capacity on Demand
- Dynamic reconfiguration of partitions
- Support for deploying plans created by the system planning tool
- Management of virtual Ethernet among partitions
- Clustering
- Concurrent firmware updates
- Hot add or remove of I/O drawers

For HMC updates and function and hardware prerequisites, refer to the following Web site:

http://www14.software.ibm.com/webapp/set2/sas/f/hmc/home.html

Integrated Virtualization Manager

The Integrated Virtualization Manager (IVM) provides HMC support on the system for entry-level IBM System p models. The IVM provides virtual resources through its Virtual I/O server capability.

Enterprise Workload Manager

Enterprise Workload Manager™ (EWLM) is a performance management tool that allows for management and monitoring of workloads. In addition, EWLM allows monitoring of application-level transactions separate from operating system processes. EWLM allows you to not only view the central processing unit (CPU) usage for systems within your domain, but also determine which work contributes the most to the overall system CPU usage.

IBM Enterprise Workload Manager (EWLM) allows you to define business-oriented performance goals for an entire domain of servers, and then provides an end-to-end view of actual performance relative to those goals. EWLM is a key component of the IBM Virtualization Engine™. EWLM offers the following features:

- Provides an understanding of the actual flow of work, based on dynamic discovery.
- Automatically detects server and application topologies so that it can rapidly and efficiently identify the likely origin of performance problems.
- Allows users to create policies that correspond to Service Level Agreements (SLAs).

- Dynamically adjusts processing units among partitions to ensure that performance goals are met.
- Communicates with load balancers to help ensure that performance goals are met.
- Encourages middleware and third-party software vendors to instrument their applications with the Open Group Application Response Measurement 4.0 (ARM) APIs. This allows EWLM to obtain granular performance data for multi-tiered application-level transactions.
- Provides formal, public programming interfaces on which systems management products can build. These products can use EWLM to handle interactions with ARM-instrumented middleware and to provide server and application topology and performance statistics.

For more information about EWLM, see *Quantifying the Benefit of EWLM in a System p Environment*, REDP-4394.

Resource Dependency Service

You can use Resource Dependency Service (RDS) to see a graphical representation of the dependencies and relationships among the physical resources, virtual resources, and applications that reside in your environment.

To show these relationships, RDS collects data about the virtual resources, physical resources, and applications within your environment. RDS then analyzes the data it collects and displays the data graphically using the topology viewer in the Virtualization Engine console. The data that the topology viewer displays shows you the resources that exist in your environment, but more importantly, it shows the relationships and dependencies among those resources.

You can use the dependency data to simplify a number of your IT management tasks. For example, you can use this data to improve your backup and recovery strategy by inferring what will happen if a disk unit, server, or application fails. In addition, you can see information such as which applications or servers are using a database. You can then infer which servers or applications are affected if you stop a database for a backup operation.

To configure the Resource Dependency Service:

1. In the navigation pane of the Virtualization Engine console, expand **Virtualization Engine Console** → **Virtualization Engine Setup** → **Resource Dependency Service Configuration**.
2. In the RDS welcome window, click **Configure data collectors** for standard resources.

3. In the Configure Data Collectors window, click **Select Action** and select **New**.
4. Click **Go**.
5. In the New Data Collector window, select **IBM Director** and click **OK**.
6. From the General page, specify a name for the Data collector field, such as MyCoDirDC.
7. Specify the host name or IP address of the server that runs the IBM Director application.
8. Click **Credentials** and specify the user ID and password for IBM Director and click **OK**.
9. Click **Schedule**.
10. Select **Do not collect** (if you do not want to immediately schedule a data collection) and click **OK**.
11. . To run a data collection right away, complete the following steps:
    a. At the Configure Data Collectors window, select the new data collector, then select **Select action** → **Collect now**, and then click **Go**.
    b. At the Collect Now window, click **Collect**.

When the IBM Director data collector completes the collection operation, the Last Run column in the When to collect window gives a status of Completed.

Systems provisioning

Systems provisioning is the supplying of server resources (typically partitions) to applications that need them. Systems provisioning may be performed at a variety of levels:

- Manually
- Using a NIM server
- Using an HMC or IVM
- Using IBM Director Virtualization Engine
- Using Tivoli Provisioning Manager

Energy management

POWER6 IBM System p servers provide energy management functions to help reduce total power consumption and improve the energy efficiency of your IT infrastructure. Power management features may be managed individually through the HMC or, for multiple servers, through the Active Energy Manager extension to IBM Systems Director.

Mid-range and high-end POWER6 servers provide a power management feature that, when enabled, lowers the processor voltage and clock frequency to reduce total power consumption. To enable power management, in the Servers view of

the HMC, select the server, and then select **Operations** → **Power Management™**. Then, in the dialog window that appears, select **Enable**. See Figure 4-8.

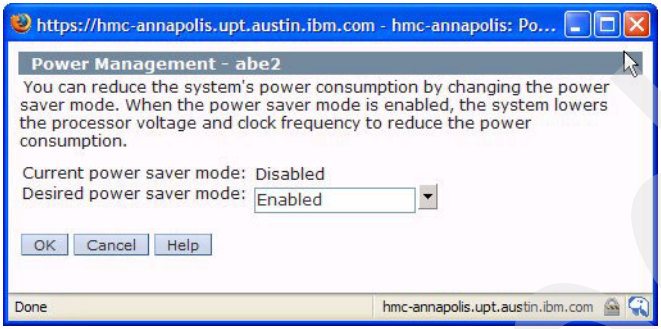


Figure 4-8 Enabling Power Management

Active Energy Manager enables centralized power monitoring and managing of servers. Active Energy Manager supports all IBM System p servers, including BladeCenters. The following are the energy management features available on Active Energy Manager:

Power Trending Displays power usage for individual systems over time in graph or table format

Thermal Trending Displays information about the inlet and exhaust temperatures for individual systems

Power Capping Throttles the processor to use less power if the system consumes more than a maximum level

Power Savings Mode Enables power savings of up to 30% of normal CPU power usage

iPDU support Enables power trending to be performed on older systems

System p integration with IBM Director

IBM Director is an integrated suite of tools that provides you with a comprehensive system management solution for heterogeneous environments, and is tightly integrated to IBM System p management functions of the HMC and IVM. IBM Director automates numerous processes that are required to manage your infrastructure proactively, including software distribution, system inventory, monitoring, remote hardware control, task execution, and more. IBM Director provides a comprehensive suite of system management capabilities. For more information about IBM Director and licensing information, refer to *IBM Director on System p5*, REDP-4219.

Virtualization Manager extension to IBM Director

Virtualization Manager is an extension to IBM Director that helps you manage both physical and virtual resources from one location. Virtualization Manager helps you optimize your virtual resources as well as plan for additional physical resources. Virtualization Manager helps you optimize your virtual resources by giving you views of relationships between your physical and virtual resources. The topology map view visually shows the relationships and dependencies between physical and virtual resources. To help with planning for additional resources, Virtualization Manager includes a link to the System Planning Tool feature of the HMC, which helps you plan and order new System p servers.

When configuring new systems, you can use the Create Virtual Server wizard within Virtualization Manager to quickly create logical partitions. With IBM Systems Director Virtual Image Management, a separately installable component of Virtualization Manager, you can create and manage images and system templates to expedite the process of creating logical partitions.

Virtualization Manager may also be used to monitor system health. Utilization levels and health status are reported and displayed in graphical form, enabling you to easily see status and systems management needs.

To summarize, Virtualization Manager offers the following capabilities:

- Manage logical partitions (virtual servers).
- Expedite the deployment of virtual servers through templates.
- View the health of resources in your environment.
- View topology maps showing physical and virtual relationships.
- Access planning tools for IBM System p servers.
- Access storage management tools.
- Access embedded Fibre Channel switch management tools.

Electronic Service Agent

Electronic Service Agent™ automatically monitors and collects hardware problem information and sends this information to IBM support. It also can collect hardware, software, system configuration, and performance management information that may help IBM support assist in diagnosing problems.

Electronic Service Agent is a no-charge software tool that resides on your system to continuously monitor events and periodically sends service information to IBM support on a user definable time table. This tool tracks and captures service information, hardware error logs, and performance information. It automatically reports hardware error information to IBM support as long as the system is under an IBM maintenance agreement or within the IBM warranty

period. Service information and performance information reporting do not require an IBM maintenance agreement or do not need to be within the IBM warranty period to be reported. Information collected through the Electronic Service Agent is available to IBM service support representatives to help them in diagnosing problems.

Concurrent maintenance

POWER6 processor-based systems are designed for redundant and concurrently maintainable power, fans, physical storage, and I/O towers.

4.3 Operating system features overview

The following sections show specific features for AIX and Linux on System p. This involves direct influences in the form of feedback towards the System p hardware. Figure 4-9 shows the features being discussed in this section and shows the escalation of benefits from the hardware to the operating system. The operating system on the other hand can give feedback to the hardware.

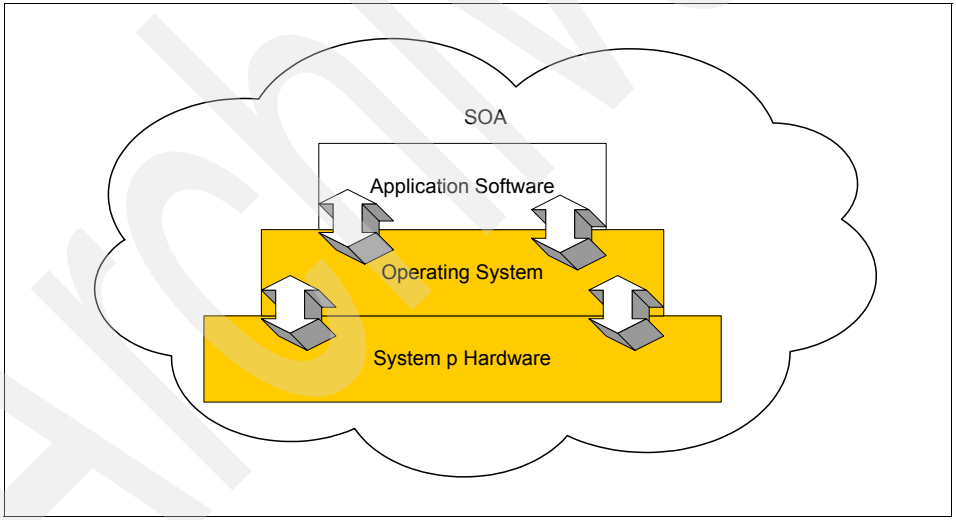


Figure 4-9 Feature overview: operating system

Within System p, the operating system can be enabled to perform certain operations on the hardware to adjust to new requirements or needs. For these operations, no specific system tool has to be used. Changes can be triggered right from the point where they are required.

4.3.1 Operating system support summary

The POWER5-based System p5™ servers and POWER6-based System p servers support the following operating systems:

- IBM AIX 5L
- IBM AIX Version 6.1
- Linux distributions from SUSE and Red Hat

IBM AIX 5L

AIX 5L is supported on the System p servers in partitions with dedicated processors and shared-processor partitions (micro-partitions). A combination of AIX 5L V5.2 ML2 and AIX 5L V5.3 partitions with dedicated processors and adapters, and of AIX 5L V5.3 partitions using Micro-Partitioning™ and virtual devices. AIX 5L V5.3 partitions can use physical and virtual resources at the same time.

IBM AIX V6.1

AIX V6.1 includes significant new capabilities for virtualization, enterprise security features, near-continuous availability, and simplified manageability.

AIX Version 6.1 runs on systems based on POWER4, PPC970, POWER5, and POWER6 processors. Most features of AIX V6.1 are available on all supported hardware. A few features are only available when AIX V6.1 is running on a system built with POWER6 processors. Table 4-1 lists selected features of AIX V6.1 and whether those features require POWER6 processors.

Table 4-1 AIX V6.1 features and platform

AIX V6.1 feature	Processor support
Workload partitions	POWER4, PPC970, POWER5, and POWER6
Live Application Mobility	POWER4, PPC970, POWER5, and POWER6
Application storage keys	POWER6 (also supported by AIX 5L V5.3)
Kernel storage keys	POWER6
Automatic variable page size	POWER6
Firmware assisted dump	POWER6
Decimal floating-point	POWER6 (also supported by AIX 5L V5.3)
Role-based access control	POWER4, PPC970, POWER5, and POWER6
File system encryption	POWER4, PPC970, POWER5, and POWER6

AIX V6.1 feature	Processor support
Trusted AIX	POWER4, PPC970, POWER5, and POWER6
probevue dynamic tracing	POWER4, PPC970, POWER5, and POWER6

AIX V6.1 only supports the 64-bit kernel. Any 32-bit and 64-bit applications that ran on AIX 5L can continue to run unchanged on AIX V6.1, but 32-bit kernel extensions and device drivers are not supported on AIX V6.1. IBM systems based on the POWER6 processor such as the IBM System p 570 provide additional virtualization capabilities of the IBM POWER VM feature that are supported by AIX 5L V5.3 as well as AIX V6.1.

Linux on System p and distributions

Linux is an open source operating system that runs on numerous platforms from embedded systems to mainframe computers. It provides a UNIX-like implementation across many computer architectures. This section discusses two versions of Linux to be run in partitions; it does not discuss the Linux-based Virtual I/O Server. The supported versions of Linux on System p servers are:

- Novell® SUSE Linux

  SUSE Linux was the first of the IBM Linux Distribution Companies to release Linux for the System p. Since then, SUSE Linux was acquired by Novell and is now called Novell SUSE Linux. For more information about Novell SUSE Linux, see:

  http://www.novell.com/linux

  For the convenience of clients, IBM will provide the ability to order a full retail distribution of SUSE Linux Enterprise Server 10 in conjunction with new System p5 server and JS21 blade purchases, processor upgrades or, as appropriate, CoD activation. IBM will accept client orders and payment for Novell SUSE Linux and deliver the code with the respective systems. Maintenance and support can also be provided for an additional charge. Clients always have the option of ordering directly from Novell SUSE Linux at any time per the information above, either through their Web site or from Novell SUSE Linux Business Partners.

  The latest version of Novell SUSE Linux for enterprise clients, SUSE Linux Enterprise Server 10 for POWER, became available in July 2006 and contains the 64-bit Linux operating system based on the 2.6.15 kernel and supports both 32- and 64-bit applications. Full details on SUSE Linux Enterprise Server 10 for System p servers and BladeCenter® JS21 blades are available directly from Novell at:

  http://www.novell.com/products/server/

- Red Hat

  Founded in 1994, Red Hat is a well known Linux provider and is the market leader as a Linux distributor. For more information about Red Hat, go to:

  http://www.redhat.com

  Red Hat Enterprise Linux AS 3 for POWER became generally available for IBM eServer pSeries in November 2003 and was updated to support IBM eServer p5 and OpenPower® servers and JS20 blades in August 2004 with Red Hat Enterprise Linux AS 3 Update 3. This is a full 64-bit kernel (based on the 2.4.21 kernel with selective code backported from the 2.6 kernel, such as the NTPL and simultaneous multithreading support) with 32- and 64-bit application support. On February 2005, Red Hat Enterprise Linux AS 4 became available from Red Hat.

  This release is based on the 2.6 kernel, which includes Large Page support and the Preemptive kernel. This version supports POWER4, POWER5, POWER5+™, and POWER6 processor-based servers and JS20/JS21 blades. The current release of Red Hat Enterprise Linux for POWER is Version 5, which is also based on 2.6 kernel.

  For the convenience of clients, IBM provides the ability to order a full distribution of Red Hat Enterprise Linux AS 4 in conjunction with new System p5 servers and 16 Virtualizing an Infrastructure with System p and Linux BladeCenter JS21 purchases, with any new processor upgrade or, as appropriate, with any activation of a Capacity on Demand (CoD) processor. IBM will accept the client's order and will have the Linux distribution arrive with the server shipment at the client location. Clients always have the option of ordering directly from Red Hat, Inc. at any time from the Red Hat Web site or a Red Hat Business Partner. Red Hat Enterprise Linux is also available in an evaluation version from Red Hat.

  You can find full information about this product, including pricing and support options, at:

  http://www.ibm.com/systems/linux/power

Consider the following items when configuring Linux partitions in virtualized System p systems:

- Not all devices and features supported by the AIX operating system are supported in logical partitions running the Linux operating system.
- Linux operating system licenses are ordered separately from the hardware. Linux operating system licenses may be acquired from IBM or from other Linux distributors.
- Regardless of how a Linux distribution is ordered, the distributors offer maintenance and support. IBM also has support offerings from IBM Global Services for these distributions.

- While Linux can be run successfully on partitions with more than eight processors (the Linux kernel 2.6 may scale up to 16 or even 24 processors for certain workloads), typical Linux workloads only effectively utilize up to four or eight processors.
- The Linux 2.6 kernel has been found to scale well up to 32 cores and in selected workloads to 64-core processors in an SMP system depending on the workload. This scaling makes it a good match for systems with 4-cores and up capabilities.

4.3.2 Virtualization features

This section describes specific features of the AIX and Linux operating systems that support the virtualization and partitioning technology of System p.

AIX supported virtualization features

The following features can be configured in AIX to adjust the hardware from inside of the operating system without needing to use the HMC.

- Logical partition support

  Logical partitions can be adjusted from inside the operating systems in terms of adding CPU resources, or adding memory or PCI/IO resources.

- Micro-partitioning support

  Micro partitions can be changed from within AIX to incorporate changes in capacity requirements.

New virtualization support in AIX V6.1 includes:

- PowerVM workload partitions (WPAR)
- PowerVM Live Application Mobility (with the IBM PowerVM Workload Partitions Manager for AIX)
- PowerVM live partition mobility

Linux supported virtualization features

We discuss the following Linux distributions and versions:

- SUSE Linux Enterprise Server 9
- SUSE Linux Enterprise Server 10
- Red Hat Enterprise Linux Advanced Server 4
- Red Hat Enterprise Linux Advanced Server 4.5
- Red Hat Enterprise Linux Advanced Server 5

The following features are available:

- Dedicated-processor partitions
- Shared-processor partitions and virtual processors, capped and uncapped
- Dynamic reconfiguration of processors (DLPAR)

The Linux distributions do not support the dynamic reconfiguration of memory and I/O slots from the Linux partitions.

4.3.3 Security features

This section introduces security from a system security viewpoint rather than a data security perspective. On IBM System p machines, all resources are controlled by the POWER Hypervisor. The POWER Hypervisor ensures that any partition attempting to access resources within the system has permission to do so. PowerVM includes a number of technologies allowing partitions to securely communicate within a physical system. To maintain the complete isolation of partition resources, the POWER Hypervisor enforces communications standards, as normally applied to external infrastructure communications. For example, the virtual Ethernet implementation is based on the IEEE 802.1Q standard.

AIX security features

Providing for a secure computing environment has always been a key goal for the AIX operating system. AIX V6.1 is designed to be compliant under the common criteria at Evaluation Assurance Level 4+ (EAL4+), including the Role Based Access Control Protection Profile (RBACPP) and the Labeled Security Protection Profile (LSPP). You can find more details at the following Web site:

`http://www-03.ibm.com/systems/p/os/aix/certifications/index.html`

It includes many new features that can increase security while reducing the effort needed to provide a secure infrastructure:

- Secure by Default

  Secure by Default is a new install function, which can be invoked as part of AIX Security Expert to do a minimal installation. Secure by Default is an even more restricted installation than the high security setting in AIX Security Expert. Secure by Default further protects systems by not allowing network programs to be active before systems are hardened. Secure by Default significantly reduces network vulnerabilities and vulnerabilities from problems in software that can be exploited by performing a minimal install without network programs. After the initial installation, the administrator only installs what is explicitly necessary on a system.

Secure by Default should be used when security is one of the major requirements, for example, within systems that store confidential information, customer accounts, or personal data, such as credit card numbers or social security numbers.

- Role Based Access Control

  Role Based Access Control (RBAC) provides improved security and manageability by allowing administrators to grant authorization for management of specific AIX V6.1 resources to users other than root. RBAC can also be used to associate specific management privileges with programs, which can reduce the need to run those programs under the root user or through setuid. RBAC improves security by reducing the number of root users required to manage systems. It can reduce administrative costs and improve administrative efficiency by allowing secure delegation of routine administrative tasks to nonroot users.

  The traditional access control model in UNIX systems is known as Discretionary Access Control (DAC). This means that the person owning the file or directory has the discretion of setting the permissions for who may read, write, or execute the file. This is the standard read-write-execute to which all UNIX users are accustomed. In order to perform privileged operations, users must either be part of the system group, or elevate their privileges to root authority. Allowing many system administrators access to the root password of a system or group of systems can be an uncomfortable situation in today's businesses. Responsibilities for various parts of the computer resources and networks is often divided among several administrators. In fact, in some cases administrators may be outside vendors who administer a select application or piece of hardware. Although it requires some planning and policy creation, using the fine-grained access control implementation methods provided by RBAC can give you a lower risk strategy than an alternative strategies where root passwords are distributed to several people.

  There are additional situations where it may be desirable to divide authorities among two or more people to require multiple people to perform certain operations. This prevents abuse of power, such as creating user accounts and setting passwords, which can be misused to damage company assets or important files, such as audit logs. Currently, the root user ID 0 is needed to perform privileged operations, so root passwords must be shared among many administrators. This can be a logistical headache when passwords need to be updated or administrators leave the company and it is not secure to have non-employees know root passwords. RBAC is enabled by default on an AIX V6 installation.

There are many advantages of the AIX V6 Role Based Access Control mechanisms. They include:

- Selective assignment of privileged access roles to system users
- More levels of granularity than previously offered
- Integration into a centralized policy infrastructure with LDAP
- Option to eliminate the root user
- Re-authentication necessary when switching roles, which reduces collaborative attacks
- Ability to define custom authorizations and roles without reprogramming applications

- Trusted AIX

  Trusted AIX is the feature that provides the framework for Mandatory Access Controls and Labeled Security. The roots of Mandatory Access Control (MAC) systems come from the Department of Defense Orange book standards developed in 1985. The labeling capability made it possible to categorize resources and subjects based on security labels. This provided the ability to create vertically defined hierarchies as well as horizontal compartmentalization to separate data and prevent unauthorized access. This system eliminates the concept of the all powerful root ID. Security policies are defined for a site-wide or institution-wide basis and cannot be circumvented by local administrators.

  Multi-level security associates subjects and objects. Subjects may be processes or users. Objects may include things like devices, network packets, files, and segments. Authorities and policies are defined well before data is populated. The concept of Mandatory Access Control is to prevent data leakage. Therefore, a user or process is defined to operate in a range of values. The objects have a certain classification level. Users may write up or read down. For example, an information gatherer can write a report that is labeled as higher than his or her level. Once the report is written, he or she can no longer access it.

  Trusted AIX extends the security capabilities of the AIX OS by integrating compartmentalized, multilevel security (MLS) into the base operating system to meet critical government and private industry security requirements. Trusted AIX is implemented as an installation option that can provide the highest levels of label based security to meet critical government and private industry security requirements. Trusted AIX supports various MLS features, such as partitioned directories, trusted networking, and labeled printing.

Trusted AIX allows for the setting of two label types: *sensitivity labels* and *integrity labels*. Objects are labeled with their level of security. Processes and users operate in a range or sensitivity levels and the objects they access must be defined within the range of security levels they are permitted to access.

Integrity levels provide a scheme for determining how a user or process can modify an object. Even if it is within range of their security label, they can be further restricted whether they can write the object, read the object, or append to the object, for example.

Trusted AIX uses the Role Based Access Control capabilities to assign privileges and authorizations. Trusted AIX's strict definitions, security controls, and integrity levels allows systems to:

- Protect servers from internal and external attackers
- Compartmentalize and secure applications
- Prevent malicious code from damaging systems
- Limit access to administrative (superuser/root) privileges
- Meet or exceed government standards for maximum security

Trusted AIX should be used when data is of different security classification levels, and where it is imperative that data is not leaked from one level to another. Additionally, it may be used to provide a mandatory access control security schema where security policies are defined in a top-down hierarchical manner.

- File Permission Manager

  Reducing the number of set UID bit programs is a best practice for hardening systems. The File Permission Manager command is designed to aid administrators in reducing the number of set UID bit programs.

  It can be used to remove UID permissions on programs if desired and provides a way of setting them back to the default setting if the program may not run as expected. This prevents possible flaws through malicious command execution of system programs or self written programs.

- Workload Partitions

  WPARs provide an isolated application environment that minimizes potential interactions with system objects or processes outside the WPAR. Even though a WPAR is using the same operating system instance as other WPARs or the global environment, the ability to view or interact with processes or devices outside the partition is limited.

In system workload partitions, the ability to interact with the global environment's file systems is limited. For non-persistent system objects, such as IPC and processes, constraints are implemented in the following ways:

- Marking system objects as being owned by an individual WPAR
- Denying users or applications in one WPAR the ability to view or interact with system objects owned by other

The features also include marking of system objects as being owned by an individual WPAR to dedicate this object. Users and applications in one WPAR can be denied viewing or interacting with system objects owned by other WPARs.

Interaction with persistent objects, such as files and devices, is also limited by maintaining separate file systems local to an individual partition. For persistent objects, constraints are implemented in the following ways:

- Creating and mounting unique file systems separate from file systems available in other WPARs
- Mounting shared, read-only file systems using the namefs mount type from the global environment
- Maintaining an environment within the local file system mounts for all processes using the **chroot** command

- Encrypting File System

  The IBM Enhanced Journaled Filesystem Extended (JFS2) adds even greater data security with the capability to encrypt the data in a file system. Clients can select from a number of different encryption algorithms. The encrypted data can even be backed up in encrypted format, reducing the risk of data being compromised if backup media is lost or stolen. The Encrypting Filesystem can even help prevent the compromise of data by root level users. The Encrypting Filesystem does not require significant additional administrative effort because the key management is automatic and fully integrated into the login authentication process.

  AIX V6 includes the capability to encrypt files in the JFS2 file system with the use of the Encrypting File System (EFS). The encrypting file system encrypts files on a per file basis. Users can either create their own key store, or use a group key store of which they are a member.

  The management of user key stores is integrated into the existing user administrator keys. The cryptographic access to files is also integrated into existing commands like **chmod**. This minimizes the system administration tasks, and reduces changes of misconfiguration errors. A few EFS specific commands have been added to allow for complete control by the system administrator.

For most operations, the use of an encrypted file is transparent to the user. The underlying commands encrypt and decrypt the data. The specific key and algorithm information for each file is saved in the file meta data, and the commands are programmed to know how to process the files. The keys used to encrypt the files are protected by an asymmetric private key.

The user keystore to access the encrypted file system is opened and loaded upon successful login. A process needing to access an encrypted file verifies the user's credentials and then is able to decrypt the file.

The Encrypting File System provides the following benefits:

- Transparent to users and system administrators.
- Increased file level encryption granularity: Many other encryption products use volume level encryption. All users using files in the volume use the same encryption keys. This type of encryption is useful for protecting the theft of a disk, but does not protect data from being read by others allowed to use the same volume.
- EFS has a unique mode that can protect against a compromised or malicious root user.
- Can assign users to groups and use group keys.
- Centralized keystore.
- Employs AES symmetric encryption algorithm. AES Key length and mode selectable by user.
- EFS is integrated into user administration commands.

Encrypting File System should be used to provide extra protection for privacy and when required by law or business practices to encrypt sensitive data. EFS requires the installation of a cryptographic toolkit.

- AIX Security Expert

The AIX Security Expert provides clients with the capability to manage more than 300 system security settings from a single interface. To configure security on a system, you start with a template that provides the initial configuration and then customize it to fit the security requirements. The Security Expert provides four templates: high, medium, or low security, or a Sarbanes Oxley template designed to help you become compliant with the security requirements of the Sarbanes Oxley Act. Once the Security Expert has been used to configure security on a system, you can export those security settings and use them to set other systems identically. With AIX V6.1, you can even store these security configurations directly in a Lightweight Directory Protocol (LDAP) directory, simplifying implementation of consistent security across an entire enterprise.

Prior to the release of AIX Security Expert in AIX 5L V5.3 TL5, security settings for AIX were distributed among a variety of system and network commands and separate SMIT panels. Creating proper default settings required expertise in many areas, and was a painstaking process. Many administrators ended up hardening their systems after installation by running home-grown scripts that evolved over time. It was difficult to ensure the compliance to settings created in these scripts. Also, if changes to the settings were made that caused undesirable behavior, there was not a good way to restore the original settings.

AIX Security Expert addresses these concerns by building a policy-based set of rules that are implemented using standard AIX commands. Users can use the graphical interface in WebSM to review a check box style of settings, generally categorized as high, medium, and low security. AIX Security Expert allows security settings to be standardized throughout the enterprise and to be invoked early in the boot process to prevent systems from being vulnerable in the network before customer scripts can be invoked. It also checks for software prerequisites and creates a message if the prerequisites are not satisfied.

AIX Security Expert for AIX V6 has been enhanced with some new features to further improve the security and prevent intrusions. These features include:

- Invoking Secure by Default for a high security setting
- Centralized Policy Distribution through Lightweight Directory Access Protocol (LDAP)
- Ability to customize and include user-defined policies
- Invocation of File Permission Manager command for managing SUID programs
- More stringent check for weak passwords

AIX Security Expert is recommended for use by all administrators. Hardening scripts can be added as additional rules and invoked by an AIX Security Expert.

- Trusted Execution

  Trusted Execution includes a group of features that verify the integrity of files in AIX. Trusted Execution provides for integrity checking of the operating system. At any point in time, an administrator can verify the state of the system by comparing the attributes of the important files in the system against a reference database. Additionally, Trusted Execution provides for monitoring of executables and kernel extensions during the load time. It can thus block any attempts to execute malicious code that is not part of the trusted database.

AIX automatically marks certain types of files as trusted, and their signatures will be calculated during installation. Hashes are calculated for runtime checks using the SHA256 hashing algorithm. SHA256 is the Secured Hash Algorithm with a 256-bit key. Additional trusted programs can be identified by the administrator as trusted programs. Those executables are signed and their signature value entered into the Trusted Signature Database (TSD). The values in the TSD are checked by the loader, by calculating the hash value of module and comparing it with the expected value stored in the database. Because even one bit change in a file will radically alter the hash value for that file, a hash is an effective mechanism to ensure the integrity of the file. Executables that do not pass the signature comparison are not permitted to load.

Trusted Execution should be used when integrity checking on customer specific files is desired. Trusted Execution is a replacement for the Trusted Computing Database.

In Trusted Execution mode, AIX V6.1 will verify the integrity of programs at execution time. This can increase security by reducing the possibility that tampered programs could be used to compromise the security of the system. A signature (SHA256/RSA) database for important system files is created automatically a part of the regular AIX V6.1 install. The Trusted Execution tool can be used to check the integrity of the system against the database. Also, the administrator can define policies such that the loads of files listed in the database are monitored and execution/loads are not allowed if the hashes do not match. Additionally, the administrator can lock the signature database or the files in the database from being modified by any one in the system, including root.

Additional AIX security capabilities

Some of the new features for V6.1 are also backported to AIX 5L V5.3, such as File Permission Manager and Long Passphrase support. Some features, such as Trusted AIX, RBAC, and Trusted Execution, are available exclusively on AIX V6.

There were some significant security functions that were released in the later releases of V5.3 that should be highlighted because of their customer value. AIX Security Expert was first released in AIX 5L V5.3 TL5, as were TCP wrappers, ipfilters, and stack execution disable.

- AIX security certifications

  IBM has invested much effort and expense in the independent assurance of security functions through the process of obtaining security certifications. The use of independent auditors with deep security skills provides for a thorough review of designs, code, and security testing of AIX.

Third-party companies are hired to review the documentation and code, write the necessary security documents, and respond to comments from the security standards bodies. Once the entire process is complete, the security certification is granted.

- CAPP and LSPP

  AIX has several key security certifications. It has been certified according to the Common Criteria's Controlled Access Protection Profile (CAPP) at level EAL4+. This certification is according to a protection profile defined by the security consortium's Common Criteria. It is an internationally recognized standard. This organization defines different security profiles, and companies can certify their products at various levels of security. AXI received the CAPP certification for AIX 5L V5.2 TL6 and AIX 5L V5.3 TL5.

  AIX and Pitbull Foundation by Argus® Systems is also certified at Labeled Security Protection Profile (LSPP). This is a very strict security profile that requires the use of Mandatory Access Controls and data labeling. Pitbull Foundation is a specially modified version of AIX to implement data labeling that is available for AIX 5L V5.2 and AIX 5L V5.3. In AIX V6.1, the Mandatory Access Control and data labeling function is built into AIX.

  At the time of the publishing of this book, AIX V6 is in the process of being certified for CAPP, LSPP, and Role Based Protection Profile (RBPP). Please contact the AIX security group with specific questions about the progress of these certifications if they are required for your environment.

  AIX V6 is different than prior operating systems where different installs were needed for CAPP and LSPP. In AIX V6, there will be an install option for whether a CAPP system or a LSPP system is desired. The LSPP system builds upon the security included in the CAPP profile, and extends it for the Labeled Security function. Because many environments may require CAPP without the extra security and configuration required of an LSPP system, there is a choice of configuring AIX V6 in CAPP or LSPP mode.

- EAL4+

  EAL4+ is the evaluation level for which AIX obtained its security certification. It means "Evaluation Assurance Level 4 Plus". The "plus" refers to an additional requirement known as "flaw remediation". This means that there is a program in place to actively respond to any reported security problems and to be able to notify and distribute security patches to customers very quickly.

Linux security features

Like AIX, Linux provides many features to reach a high level of security on an operating system level. There are many different features and kernel additions that contribute to a higher level of Linux security, and it is out of the scope of the book to name all of them.

We limit our list to some more common features and support capabilities of the Linux versions supported by System p that also may relate to features mentioned for AIX.

- Pluggable Authentication Modules

  Pluggable Authentication Modules (PAM) is a system security tool that allows system administrators to set authentication policies without having to recompile programs that do authentication. You can discover more about the Linux-PAM project at:

  http://www.kernel.org/pub/linux/libs/pam/index.html

- Security Enhanced Linux on Red Hat

  Security Enhanced Linux (SELinux) is an open source research project sponsored by the National Security Agency (NSA) to provide mandatory access control in Linux. SELinux provides several benefits. By leveraging the principle of least privilege and through the institution of a security policy on the system, SELinux prevents the compromise of an entire system due to the compromise of a single application running with what would otherwise be elevated privileges. Programs are placed into individual sandboxes, isolating them from one another and from the underlying operating system.

- Red Hat Network

  This is a Red Hat service that provides simple tools to efficiently manage the life cycle of systems on your network, including provisioning new systems, managing their updates and configuration changes, monitoring system performance, and eventually re-deploying the systems for a new purpose.

- Novell AppArmor® on SUSE

  AppArmor is a network application security and provides mandatory access control for programs, protecting against the exploitation of software flaws and compromised systems. AppArmor includes everything you need to provide effective containment for programs (including those that run as root) to thwart attempted exploits and even zero-day attacks. AppArmor offers an advanced toolset that largely automates the development of per-program application security so that no new expertise is required.

- PowerVM Lx86 security features

  PowerVM Lx86 attempts to unify the definitions of users, groups, and passwords by collecting information from both the native POWER system and the x86 World and producing a merged view of the /etc/passwd, /etc/group, and /etc/shadow files.

More operational details and options are located in the *PowerVM Lx86 Administrator's Guide*, found at:

https://www-304.ibm.com/jct09002c/partnerworld/wps/static/pdf/systempave041706.pdf

4.3.4 Availability features

This section discusses availability features for AIX and Linux operating under System p.

AIX availability features

Over the years, the AIX OS has included many reliability features inspired by IBM mainframe technologies. The release of AIX V6.1 introduces many new availability features to the UNIX market that can help reduce planned and unplanned outages. New AIX V6.1 availability features include:

- Concurrent AIX kernel updates

  Concurrent AIX updates provide a new capability to deliver some kernel updates as interim fixes that will not require a system reboot to put into effect. This can reduce the number of unplanned outages required to maintain a secure, reliable system.

- Kernel support for POWER6 storage keys

  This AIX V6.1 feature brings a mainframe-inspired reliability capability to the UNIX market for the first time. Enabled by the POWER6 processor, Storage Keys can reduce the number of intermittent outages associated with undetected memory overlays inside the AIX kernel and kernel extensions. Applications can also use the POWER6 Storage Keys feature to increase the reliability of large, complex applications running under the AIX 5L V5.3 or AIX V6.1 releases.

- Dynamic tracing

  AIX V6.1 provides a new dynamic tracing capability that can simplify debugging complex system or application code. This dynamic tracing facility will be introduced through a new tracing command, **probevue**, which allows a developer or system administrator to dynamically place probes in existing application or kernel code, without requiring special source code or even recompilation. **probevue** is very flexible, allowing dynamic specification of the data to be captured at probe points and providing the ability to associate execution pre-conditions with a given probe.

- Non-intrusive service aids

  AIX V6.1 serviceability aids are designed to minimally impact performance and availability. Second Failure Data Capture (SFDC) technology involves building highly tunable diagnostic and data capture features into the operating system, but only enabling them after problem diagnosis has started. The result is faster, less-disruptive problem determination, without the need to install special “debug” code. AIX V6.1 also introduces a mainframe-inspired live dump facility that allows selected subsystems to dump their diagnostic information for subsequent service analysis, without requiring a full system dump and partition outage. For those problems that still require a partition restart in order to recover, AIX V6.1 provides a firmware-assisted dump mode on systems based on POWER6 processor technology. In this new mode, AIX V6.1 cooperates with system firmware to write the First Failure Data Capture (FFDC) information to the dump device using the restarted AIX V6.1 image, rather than writing to the dump device at the time of the failure. The result is fewer dump failures, which can enable quicker problem determination and resolution.

- Enhanced software First Failure Data Capture

  First Failure Data Capture (FFDC) gathers diagnostic information about a problem at the time the problem occurs. This removes the need to recreate the problem and reduces the time to generate diagnostic information. AIX V6.1 builds on the FFDC capabilities introduced in previous AIX releases by introducing more instrumentation to provide real time diagnostic information.

Linux availability features

Linux runs on IBM POWER6 and POWER5+ processor-based servers. It offers the enterprise class capabilities and energy efficiency that UNIX clients have depended on to run their businesses. Linux on IBM POWER6 and POWER5+ supports Capacity Upgrade on Demand (optional on selected platforms) to help meet ongoing capacity needs.

IBM now offers PowerVM Lx86 feature for all PowerVM Editions at no additional charge. You can run x86 applications to be dynamically executed on System p platforms. This significantly expands the potential application base. The ability to use x86 Linux development environment and methodology reduces the implementation risk. A single source and binary for Linux in both x86 and POWER architecture helps reduce development and maintenance costs.

A key attribute of Linux on POWER is its mission-critical reliability, availability, and serviceability (RAS) features. Drawing from autonomic computing efforts from IBM, System p servers continue to enhance the scope of their RAS capabilities.

As System p servers become capable of hosting more system images for server consolidation, the importance of isolating and handling outages that might occur becomes greater. Hardware and operating system functions have been integrated into the system design to monitor system operation, predict where outages can occur, isolate outage conditions that do occur, handle the outage condition, and when possible, continue operation. IBM RAS engineers are constantly improving the server design to help ensure that System p servers support high levels of concurrent error detection, fault isolation, recovery, and availability.

RAS capabilities have been included in Linux by IBM and the open source community for Linux on POWER. Here are some RAS features that are available on System p when running an Linux operating system:

- Chipkill and ECC memory
- Disk mirroring (software level)
- Journaled file system (several available under Linux)
- PCI extended error detection
- Redundant, hot-plug power and cooling (where available)
- Error reporting to Service Focal Point
- Error log analysis
- Boot-time processor and memory deallocation
- First Failure Data Capture
- Service Processor

Some of the RAS features are currently supported only with the Linux 2.6 kernel on POWER-based systems include:

- Hot-swapping of disk drives
- Dynamic Processor Deallocation
- Hot-plug PCI disk
- PCI extended error recovery (device driver dependent)
- Dynamic Memory Add (SUSE Linux Enterprise Server 10 only)

To support some of these features, you need to install the service and productivity tools for Linux on POWER systems. Table 4-2 on page 143 shows RAS aspects comparison for the following environments:

- Linux on IBM System p (Linux on Power)
- Linux on x86 systems (Linux on x86)
- Linux on x86 virtual environment for IBM System p (Linux on PowerVM Lx86)

Table 4-2 RAS aspect comparison

Reliability, Availability, and Serviceability features	Linux on System p	Linux on X86 system	x86 Linux on System p AVE	Comments
Automatic First-Failure Data Capture and diagnostic fault isolation	Yes	No	Yes	Used by Error Log Analysis tool
Self-healing internal POWER5 processor array redundancy	Yes	No	Yes	ECC, bit steering, memory scrubbing, and so on
Industry-first PCI bus parity error recovery	Limited	No	Limited	EEH detection: partition down versus system
Scrubbing and redundant bit-steering for self-healing in main storage	Yes	Limited	Yes	
ECC and Chipkill correction in main storage	Yes	Yes	Yes	
Fault tolerance with N+1 redundancy, dual line cords, and concurrent maintenance for power/cooling	Yes	Yes	Yes	
Predictive failure analysis on processors, caches, memory, I/O, and DASD	Yes	Limited	Yes	
Processor run time and boot time deallocation based on runtime errors	Yes	No	Yes	FFDC advantage
Fault avoidance through highly reliable component selection, component minimization, and error mitigation technology internal chips	Yes	No	Yes	
Concurrent runtime diagnostics based on First-Failure Data Capture for power, cooling, and I/O	Limited	No	Limited	
Service Processor is a separate, independent processor that provides hardware initialization during system IPL, and operation monitoring of environmental and error events	Yes	Limited	Yes	

Service and productivity tools for Linux on POWER

There are a number of additional Linux on POWER utilities that you need to install to support specific functions of a running Linux operating system on a POWER based system, including hardware service diagnostic aids and productivity tools, as well as installation aids. Some tools those are available for Linux on POWER include:

- The librtas package contains a library that allows applications to access certain functionality provided by platform firmware. This functionality is required by many of the other higher-level service and productivity tools.
- System Resource Controller (SRC) is a facility for managing daemons on a system. It provides a standard command interface for defining, unedifying, starting, stopping, querying status, and controlling trace for daemons.
- The Reliable, Scalable, Cluster Technology (RSCT) packages provide the Resource Monitoring Control (RMC) functions and infrastructure needed to monitor and manage one or more Linux systems. RMC provides a flexible and extensible system for monitoring numerous aspects of the system. It also allows customized responses to detected events.
- Cluster Systems Management (CSM) packages provide for the exchange of host-based authentication security keys. These tools also set up distributed RMC features on the HMC.

You can download these utilities and more as RPM packages from the Service and productivity tools Web site at:

https://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html

You can either download the packages to a local client and then FTP them to the Linux system, or you can download them directly to the Linux system.

Note: You need to install these additional packages regardless of whether you installed the SUSE or Red Hat distribution of Linux.

You can get more information about the tools and the installation steps in *Virtualizing an Infrastructure with System p and Linux*, SG24-7499.

4.3.5 Manageability features

This section discusses some management related functions in the AIX and Linux operating system.

AIX manageability features

AIX V6.1 also includes additional features specifically intended to improve the manageability of the AIX operating systems, such as:

- Network Installation Manager

  Network Installation Manager (NIM) provides an environment to install and manage AIX filesets on machines over the network. NIM provides an easy and efficient way for you to perform various installation and software maintenance tasks over several network types (such as Ethernet). In AIX V6, Network Installation Manager (NIM) has been enhanced to provide additional security features and flexibility by enabling the use of NFS Version 4. NIM can use NVSv4 to provide stronger, Kerberos-based security during the installation of AIX V6.1 and other software.

- AIX Workload Manager

  AIX Workload Manager allows multiple workloads to run under one AIX instance. The system administrator builds rules based upon a user, process, or workload. Based upon these rules, shares of CPU and memory adjust to the workload with peak demand. See *Introduction to Workload Partition Management in IBM AIX Version 6.1*, SG24-7431. Workload manager is also relevant for the workload partition (WPAR) concept and partition mobility.

- Management Edition for AIX

  Management Edition for AIX is an integrated systems management offering created specifically for the System p platform that provides:

  – Monitoring of the health and availability of the System p platform using IBM Tivoli Monitoring V6.2 technology

  – Discovery of configurations and relationships between System p service and application components based on Tivoli Application Dependency Discovery Manager V7.1

  – Usage and accounting of System p IT resources based on Tivoli Usage and Accounting Manager V7.1

  Refer to the following Web site:

  `http://www-03.ibm.com/systems/p/os/aix/sysmgmt/me/index.html`

- Name resolver caching daemon

  This daemon caches requests to resolve a host name, service, or netgroup to improve the efficiency of subsequent requests for the same information. Use of this facility can dramatically improve the performance of applications that are dependent on repeated requests for name resolution.

Linux manageability features

Linux support is readily available from many sources. It ranges from free support from the open source community at large (normally through e-mail or Web sites), to fee-based service contracts with service organizations and Linux distributors, such as Linuxcare, Red Hat, Inc., and Novell SUSE Linux.

Maintenance contracts for software upgrades can also be obtained from the distributors. Initial installation and maintenance is usually bundled into a Linux distributor's product.

IBM Global Services and IBM Technical Education Services has developed a comprehensive portfolio of Linux services, support and education offerings. These offerings currently include:

- Consulting, planning and implementation services:
  - Open Source Consulting
  - Linux Server Consolidation Services
  - Migration Services for Linux
  - IBM Middleware Enablement Services for Linux
  - Linux Cluster Implementation Services
- Worldwide remote 24x7 technical support:
  - 270 Support Line
  - 271 Advanced Support

Local IBM Global Services consultants are available to help clients evaluate their Linux requirements and to assist in implementing and optimizing their Linux solutions. For further details, visit the IBM Global Services Web site at:

http://www.ibm.com/services/

Linux clustering

The Beowulf clustering technology (http://www.beowulf.org) and other open source and some commercial products can be used to cluster POWER processor-based systems running Linux to provide compute or high-availability clusters. Myricom (http://www.myri.com/) has the Myrinet switch available for Linux on most System p5 servers. IBM has also announced the InfiniBand® switch, which is also supported by the Linux distributions. The Linux distributions that support the respective technologies also support the Myrinet switch and InfiniBand, specifically SLES 8 for POWER4 processor-based systems, SLES 10

for POWER5+ and POWER5 processor-based systems, RHEL AS 5 for POWER4 or POWER5 processor-based system, and RHEL AS 5 for POWER5+ and POWRER6 processor-based systems. The Myrinet switch is also supported on JS20/JS21 blade servers selectively by SLES 9 and SLES 10 RHEL AS 4 and AS 5. It can be used as a high-speed interconnect to cluster systems of System p servers running Linux. Gigabit or 10/100 Ethernet connections can also be used.

In the high availability arena, IBM is providing Tivoli System Automation and High Availability Cluster Multiprocessing (HACMP™) V5.3 and above as solutions. These products are based on technology from IBM mainframe z/OS and AIX 5L high-availability products. Other third-party solutions are also available today as well as open source solutions, such as HA-Linux. The current Novell SUSE Linux product also has built-in HA capabilities.

4.4 Realization of System p benefits to SOA

As servers support the operations of any IT infrastructure, their reliability and efficiency directly affects the business operations. Based on the unique combination System p provides, many features are specifically useful for building and supporting a SOA.

As Figure 4-10 shows, most of the features are being provided by the hardware and operating system but affect the SOA directly.

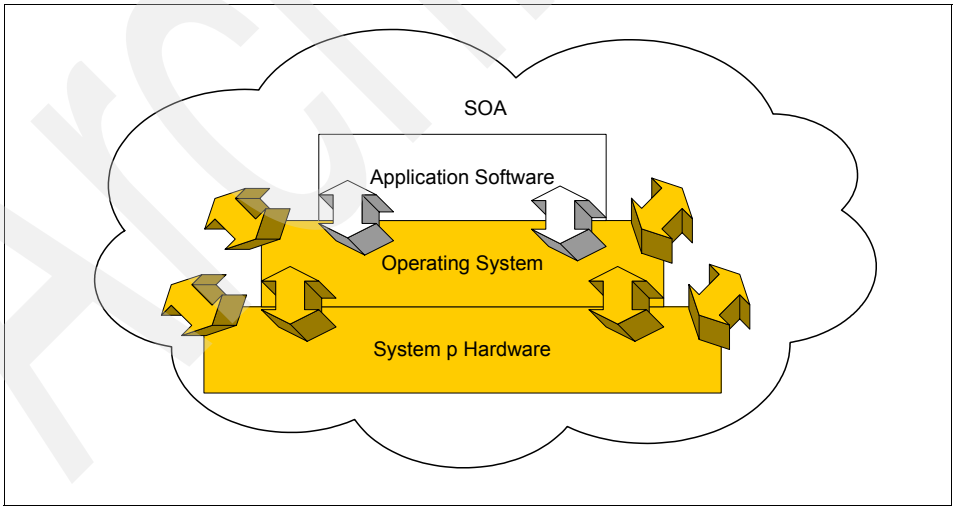


Figure 4-10 Feature overview: benefits to SOA

The SOA operational systems layer introduced in Chapter 1, “Power Systems and SOA synergy” on page 1 refers mainly to the IT infrastructure: the hardware, networks, and storage. These components support the SOA layers above by providing support for the operation of service components, middleware, and representation. The requirements of these systems vary based on the configuration, scale, and capabilities of the SOA and IT.

We see the following attributes as primary requirements for supporting a SOA:

- 4.4.1, “Flexible and scalable” on page 148
- 4.4.2, “Highly available and reliable” on page 156
- 4.4.3, “Secure” on page 160
- 4.4.4, “Optimized for reduced costs” on page 161
- 4.4.5, “Simplified management” on page 163

4.4.1 Flexible and scalable

Businesses embrace SOA so that they can create and deliver new services quickly to address changing business or customer environments. This dynamic environment can be worrisome for IT administrators. They have to be flexible to accommodate changing business load and scalable enough to serve any demand that the business requires.

Flexibility and scalability can be thought as the ability to adjust system performance based on the demand for resources. Flexibility requires changes in hardware configuration, such as the need to add more servers to cope with additional applications, while scalability concerns the ability of the existing infrastructure to cope with additional demand. These changes and extension can be served on System p servers based on the following:

- “System p virtualization benefits for SOA” on page 148
- “Deployment of virtual servers” on page 151
- “Workload partitioning” on page 153

System p virtualization benefits for SOA

Using the mentioned virtualization capabilities, System p is able to provide a unique combination of features towards the SOA operational systems layer in the SOA solution view shown in 2.5.2, “SOA solution stack and reference architecture” on page 48.

Virtualization allows greater flexibility in capacity planning, and thus provides a highly flexible layer to react to new requirements. Determining proper capacity requirements in a complex SOA environment requires careful analysis. In addition, sometimes new services are developed without having historical data to use as a basis for projecting resource requirements. Furthermore, as business needs to change or expand, additional capacity requirements are constantly being added. PowerVM's virtualization allows for additional capacities to be added for little incremental costs. Increased scalability is achieved by enabling scale adjustments without changing the physical resource configuration.

Partitioning

Micro-partitioning, as discussed in “Processor virtualization” on page 108, allows flexible processor sharing. This can be used to adjust services based on their importance. Some processor bound services with lower priority, such as maintenance functions or number crunching, may be allocated on partitions with lower processor capacity while providing real-time services with more processing power from the same shared processor pool. While there is unused processing power in the pool, the low priority services can still used the idle capacity.

The Capacity on Demand feature allows reserved capacity to be available for partitions to cope with seasonal demands without changing any of the system configuration. Capacity can be activated using a simple system reconfiguration.

Dynamic logical partitioning allows hardware resources to be moved across a partition, ensuring system availability. Partitions do not have to be restarted for a simple deallocation of faulty resources, such as a spare drive or optical disk.

Virtual networking

The use of virtual networking based on the Virtual I/O Server feature eases internal server communication and management. Services residing on a partition can easily communicate to other partitions with different services. Communication paths can be unified by limiting partition to partition communication completely in the virtual Ethernet adapter. Because routing is only necessary if network adapters are part of a different network, network adapters of one network can be used to avoid the communication through an external router just to contact another service on the same System p.

Even though this may require a detailed analysis of the network communication and some reconfiguration, it can increase performance of the service interactions and reduce load on the backbone network infrastructure. Figure 4-11 illustrates this concept.

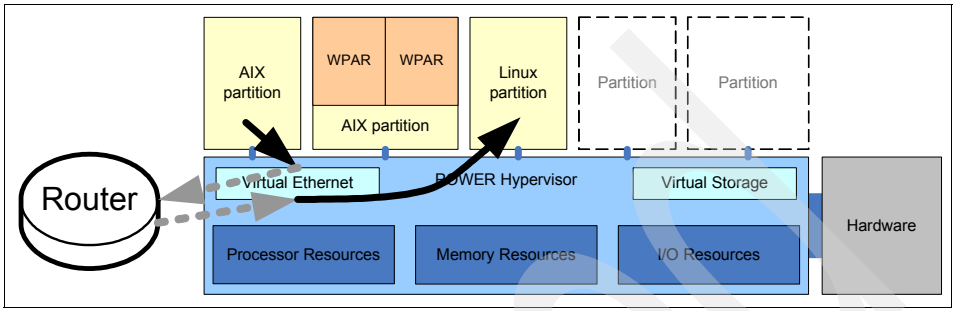


Figure 4-11 Virtual networking to avoid router processing

Virtual networking between partitions leads to a reduction of network traffic and network management. Other benefits include an increased reliability of the network infrastructure the services rely on. There are no major dependencies on external hardware or maintenance windows that may interfere with the operational status of services residing in the same system.

Virtual I/O benefits to SOA

Many services may share data or access information related to data that is stored at the same place. Partitions can access the same physical adapter through multiple virtual adapters. This feature empowers services to reside on different partitions but share the same codebase. This can also prevent issues with the different levels of software or application code, as the access is not distributed across system boundaries.

Mobility

The mobility features of System p, as discussed in “Live Partition Mobility” on page 106 and “Workload Partitions (WPAR)” on page 107, allow for an extremely high grade of flexibility. It allows the application environment to be moved to a completely different server without interrupting availability. This may be used on occasions when you have to perform emergency maintenance on one of the servers. See Figure 4-12 on page 151.

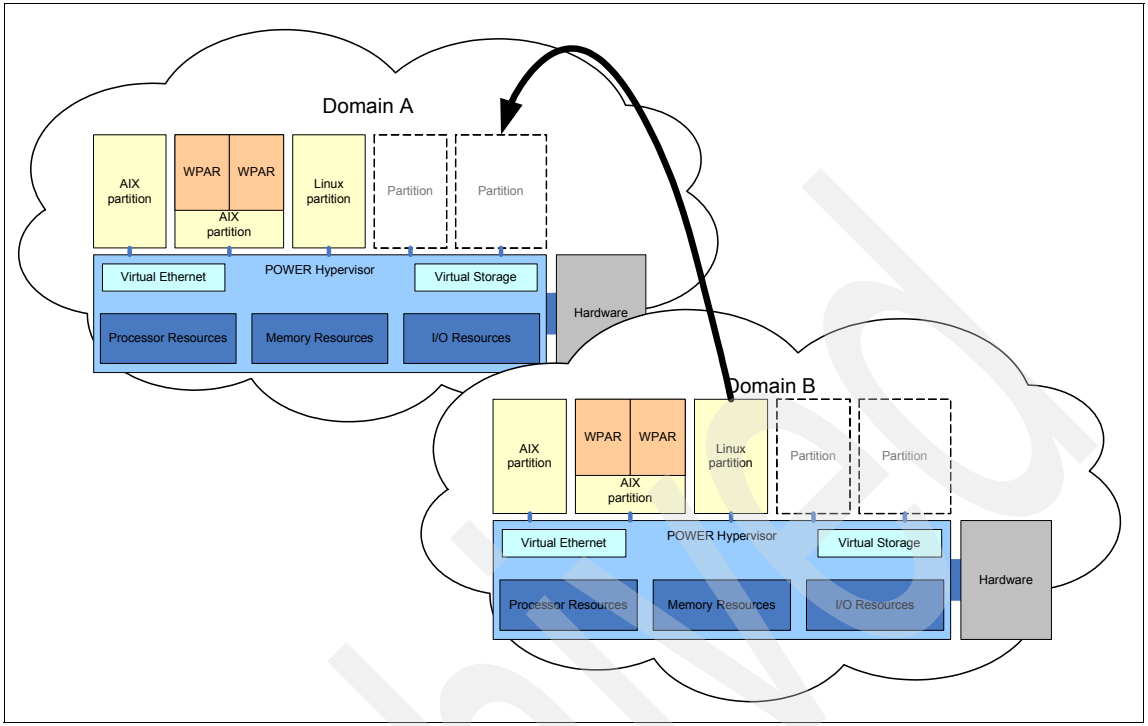


Figure 4-12 Movement of partitions across servers

This increases application availability that is independent from server maintenance:

- It allows some outages to be avoided by moving the application off of a system that needs to be shut down for maintenance.
- It can be used to balance workloads across several systems automatically or manually.
- It can be used to move workloads off servers during non-peak periods so that those servers could be turned off and therefore saving energy.

Live Application Mobility is a feature of AIX V6.1 and the Workload Partition Manager and can be used on POWER6 processors.

Deployment of virtual servers

When the need arises to deploy new servers, virtualization makes the deployment of virtual servers much easier. New virtual servers can be provisioned from existing hardware as a partition. This deployment can be performed using the Virtualization Manager.

The virtual servers deployment are performed using templates and plans. Figure 4-13 shows the relationship of these entities.

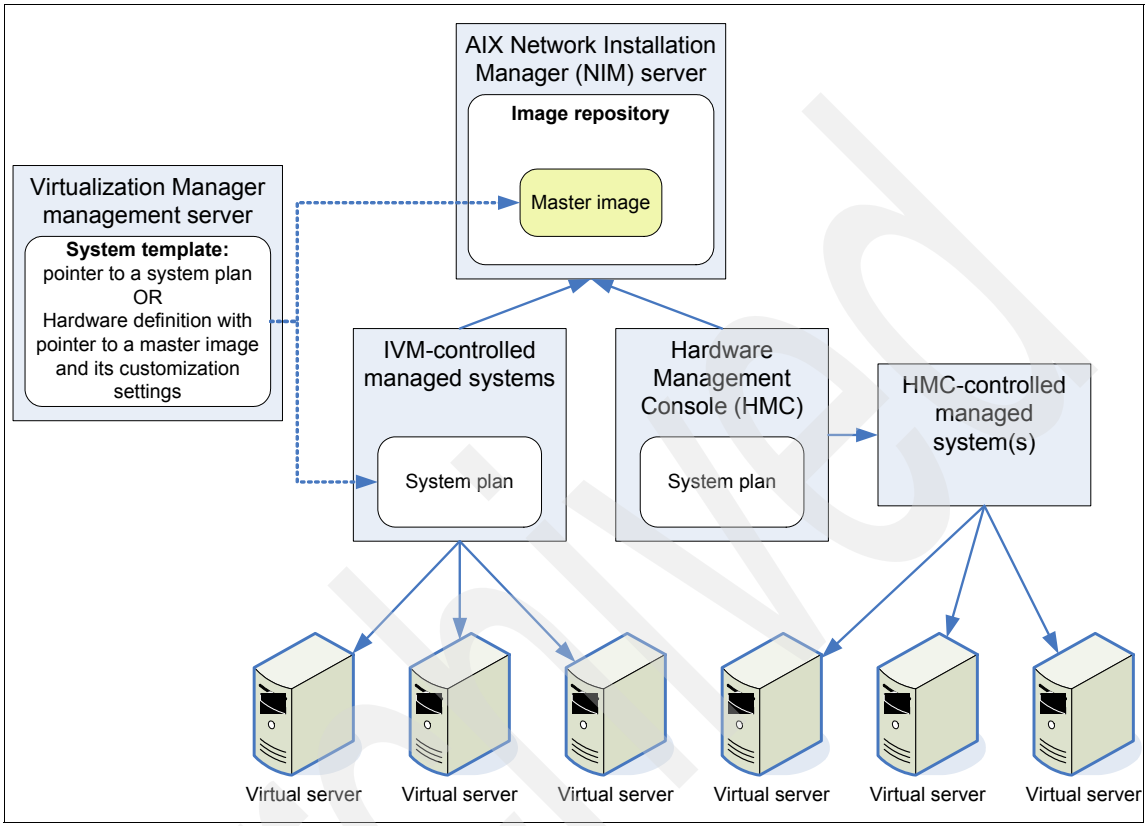


Figure 4-13 Deployment of virtual servers through Virtualization Manager

Where:

System template The system template is a hardware definition (which might include processor and memory settings), a pointer to a master image, or image customization settings such as TCP/IP settings. It can also be simply a pointer to a system plan. A system template is stored on the Virtualization Manager management server.

Master image A master image is a bootable operating system and additional software in the form of a single raw image file. The master image is stored in an Image Repository on a Network Installation Manager (NIM) server.

Image repository	An image repository is a collection of NIM **mksysb** resources. It typically resides in a NIM server.
System plan	A system plan is stored on an HMC or IVM-controlled managed system.
Virtual server	A virtual server is a logical partition (LPAR). Depending upon what the system template points to, a virtual server is based on either an image in the image repository from the NIM server or a system plan.

On IBM System p servers, you can create a system template that contains a pointer to a system plan and a hardware definition that can be transferred to one or more virtual servers that you create.

By creating a library of system templates, you are ready to deploy a new virtual server that meets specific requirements. System templates are created using the Create System Template wizard in the Virtualization Manager Web interface. A system template is deployed using the Create Virtual Server wizard. A system template can be used to quickly create multiple virtual servers based on that system template.

Workload partitioning

Workload partitioning is a mechanism to partition, not your system, but your workload (or processes). It allows multiple logically partitioned sets of processes to share a common operating system image. This facility is available on AIX.

You can partition your workload and place WebSphere Application Server processes from different services on different workload partitions. New servers can be defined and deployed on new or existing workload partitions.

Some possible uses and benefits of workload partitions include the following:

- You can consolidate multiple workloads to one AIX image yet maintain the security and isolation for each workload. You may have several development environments that all use the same application software and level of AIX but do not require their own physical LPAR. Using WPARs, you could build one LPAR, then deploy several WPARs for each development stream without needing separate hardware resources and images of AIX.
- Application software may be installed in the global environment, allowing new partitions to be quickly created without having to re-install and configure the software.
- Maintenance is simplified since there are fewer AIX instances required to maintain and upgrade.

- Administration of environments is more flexible since root privileges within WPARs are contained within the WPAR; that is, they do not affect either the global environment or other WPARs.

Workload partitions allow for greater flexibility for your operating system images. There are multiple ways to install and use applications within workload partitions. You can quickly create new isolated WebSphere Application Server environments by having the WebSphere software installed into the global environment. With the application in the global environment, new workload partitions of WebSphere Application Server may be created by simply executing a few commands. Refer to the following Technote for more information at:

http://www-1.ibm.com/support/docview.wss?rs=180&context=SSEQTP&dc=DB520&dc=DB560&uid=swg21291889&loc=en_US&cs=UTF-8&lang=en&rss=ct180websphere

Workload Partitions can be used to minimize administrative impact when consolidating systems by reducing the number of AIX instances that have to be managed. For example, instead of applying patches to multiple copies of AIX V6.1 using WPARs, you can patch the global instance, and all Workload Partitions inherit that same patch level. This helps manage growth by allowing administrators to concentrate on managing applications instead of spending time on repetitive administration tasks. Each Workload Partition can be separately administered from other WPARs in the system. The root user for a Workload Partition cannot take actions that would affect the global instance or other Workload Partitions. This isolation provides for further savings through delegation of administrative work. Workload Partitions share a single AIX V6.1 instance, so there is less isolation than there is with logical partitions (LPAR) in which each LPAR has its own independent copy of AIX V6.1. WPARs provide enough isolation for many workloads while saving administrative effort. WPARs can be used inside of LPARs, allowing the combination of the two technologies to leverage the superior isolation of LPARs with the administrative ease of WPARs.

Capacity on Demand (CoD)

To activate a Capacity on Demand feature, follow these steps:

1. From the Servers view of the HMC, select the task **View Code Information** from one of the Capacity on Demand features. A window appears that contains the required information for the activation form. See Figure 4-14 on page 155.

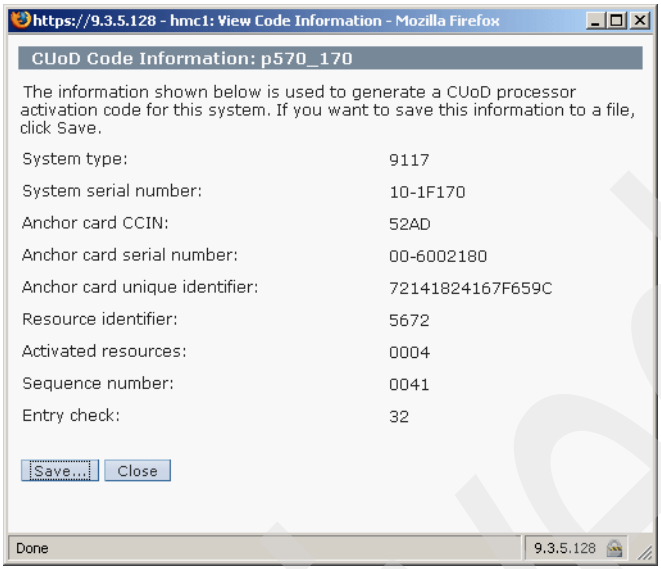


Figure 4-14 Capacity on Demand code information

2. Enter the Code Information at the Capacity in the Demand activation form on the IBM Web site found at:

   `http://www-03.ibm.com/systems/p/advantages/cod/contact/vpd_form.html`

3. Once you receive the code, in the HMC, select **Capacity on Demand (CoD)** → **Enter CoD Code** for the designated server and enter the Capacity on Demand code that is returned by the Web form.

The new capacity shows in the Capacity on Demand view of the HMC. See Figure 4-15 for an example view of the Capacity on Demand settings.

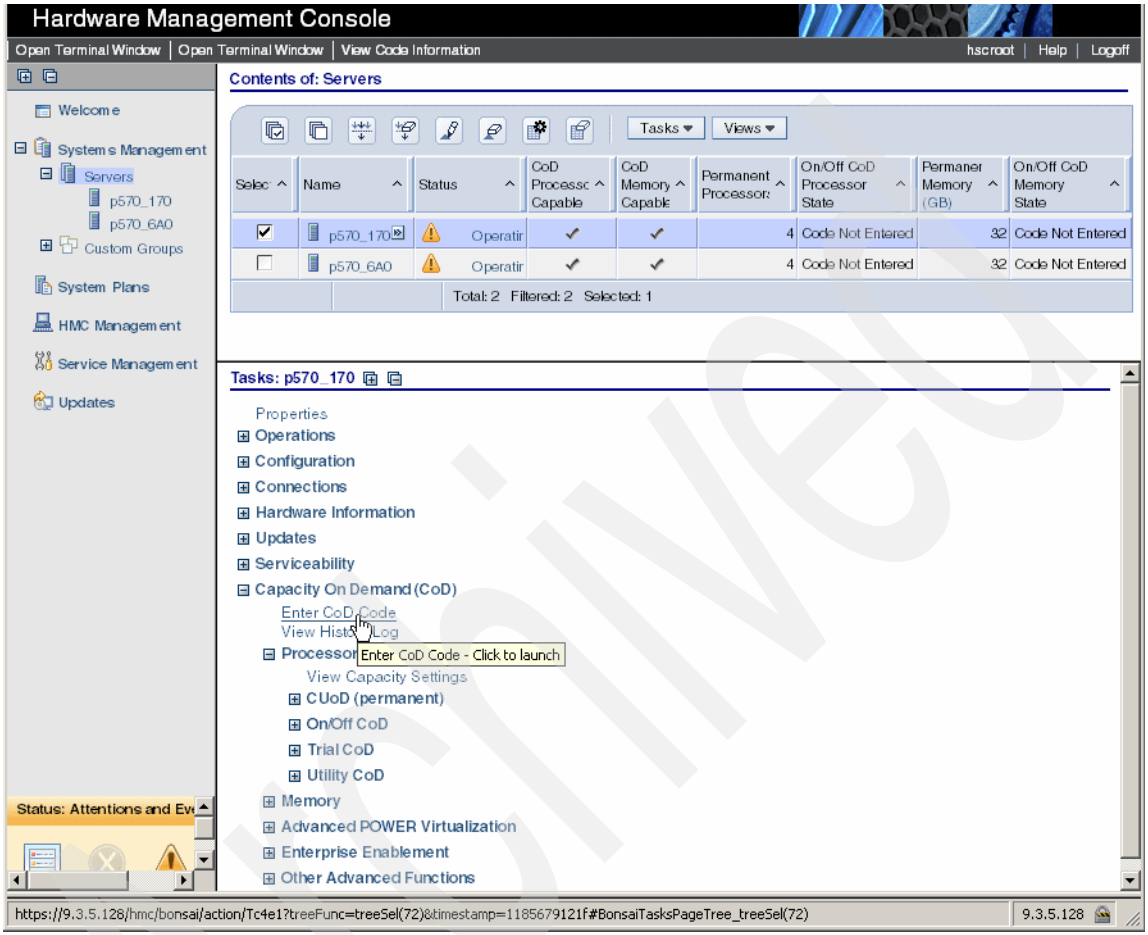


Figure 4-15 Capacity on Demand view in HMC

4.4.2 Highly available and reliable

The distributed nature of SOA applications indicates that the service consumer typically assumes that the service provider is available. The service provider's availability becomes extremely important to ensure the overall application availability.

Availability on System p servers is ensured by both hardware and operating system levels, as discussed in 4.2.3, “System p availability features” on page 116, 4.3.4, “Availability features” on page 140, and “Linux availability features” on page 141.

System p provides self recovery and redundancy built into the hardware design. It allows servers to provide resiliency. Non-critical errors can be eliminated, and even recovery from some critical errors are available. Maintenance on the system does not always generate downtime. AIX provides concurrent firmware updates. Application or even the whole partition can be moved to a different system using the partition mobility or application mobility without downtime.

For more information about the continuous availability of System p systems, refer to the following white papers:

- *IBM System p5: a Highly Available Design for Business-Critical Applications*, found at:

  http://www.ibm.com/servers/eserver/pseries/library/wp_lit.html

- *IBM POWER6 Processor-based Systems: Designed for Availability*, found at:

  http://www-03.ibm.com/systems/p/hardware/whitepapers/power6_availability.html

For information about service and productivity tools for Linux on POWER, refer to:

https://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html

Serviceability deals with how well faults and their impacts are communicated to the users and seeing how efficiently the faults are repaired with the least impact and cost. The goal of serviceability is to enable repair of the system while minimizing or eliminating service costs. IBM System p serviceability features include:

- Easier system installation
- Automated or guided service events
- System maintenance and repair
- Automatic diagnostics
- Call home

Additional procedures exist for ensuring availability. These additional processes can be achieved using software automation.

IBM Tivoli System Automation for Multiplatform

Tivoli System Automation for Multiplatform manages the availability of applications running in systems or clusters on various IBM platforms, including AIX and Linux on POWER, and offers the following features:

- High availability and resource monitoring
- Policy based automation
- Automatic recovery
- Automatic movement of applications
- Resource grouping

General product information can be obtained at the following Web page:

http://www.ibm.com/software/tivoli/products/sys-auto-linux/

You can also refer to *End-to-end Automation with IBM Tivoli System Automation for Multiplatforms*, SG24-7117.

While this is true for most areas of operations management, Tivoli System Automation for Multiplatform concentrates on automating the availability of IT resources. In other words, it provides the framework for high availability solutions, as shown in Figure 4-16 on page 159.

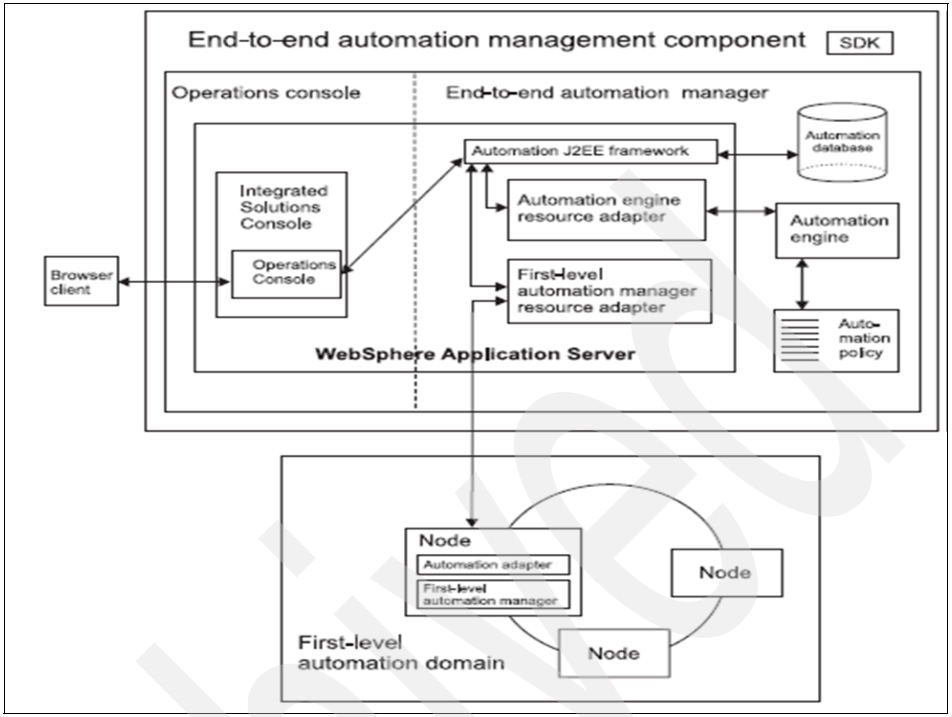


Figure 4-16 Tivoli System Automation structure

Tivoli System Automation for Multiplatforms consists of:

- Web-based user interface.
- The automation management component, which consists of the operation console application and automation manager. Automation infrastructure runs in a WebSphere Application Server. It contains console interfaces, the automation engine itself, and a repository for automation data such as policies.
- A first level automation domain. This is the automation target. It is mainly a single node or a group of nodes running the same operating system. The interface to the end-to-end automation manager is established through an automation adapter.

SteelEye LifeKeeper

SteelEye® is a data and application availability management solution for business continuity and disaster recovery on Linux and Windows. For information about SteelEye, refer to the following Web page:

http://www.steeleye.com/

The SteelEye LifeKeeper® family of application-focused data replication, high availability clustering, and disaster recovery solutions are easy to deploy and operate, and enable enterprises of all sizes to ensure continuous availability of business-critical applications, servers, and data. The solutions are proven in the most demanding of environments and are integrated to deliver flexibility, scalability, and a fast return on investment.

SteelEye LifeKeeper offers enterprise-grade reliability while simplifying implementation with certified solutions for a wide range of applications and databases running on Windows and Linux, including mySAP™, Exchange, Apache, Oracle, DB2, SQL Server, MySQL™, PostgreSQL, and others.

To complement its software solutions, SteelEye also provides a full range of high availability consulting and professional services to assist organizations with the assessment, design, and implementation of solutions for ensuring high availability within their environments.

4.4.3 Secure

Network communication inherently has many potential security vulnerabilities. Communication over an open wire can be tampered with and eavesdropped on. Security is inherently very important for a SOA application. The security must be ensured at the hardware, operating system, and software levels.

Hardware-wise, System p has the security aspects described in 4.3.3, “Security features” on page 130. AIX has been certified at Evaluation Assurance Level 4+ (EAL4+), including the Role Based Access Control Protection Profile (RBACPP) and the Labeled Security Protection Profile (LSPP).

Additional security protection on the application level can be provided by a Tivoli security solution, such as the combination of Tivoli Access Manager and Tivoli Identity Manager.

4.4.4 Optimized for reduced costs

Using a virtualized environment with server consolidation can potentially save costs. One significant saving is the ability to consolidate processing workload across different separate servers into a single pool of processing with micro-partitioning. The resulting total CPU usage would show some savings over the separate servers implementation, as shown in Figure 4-17.

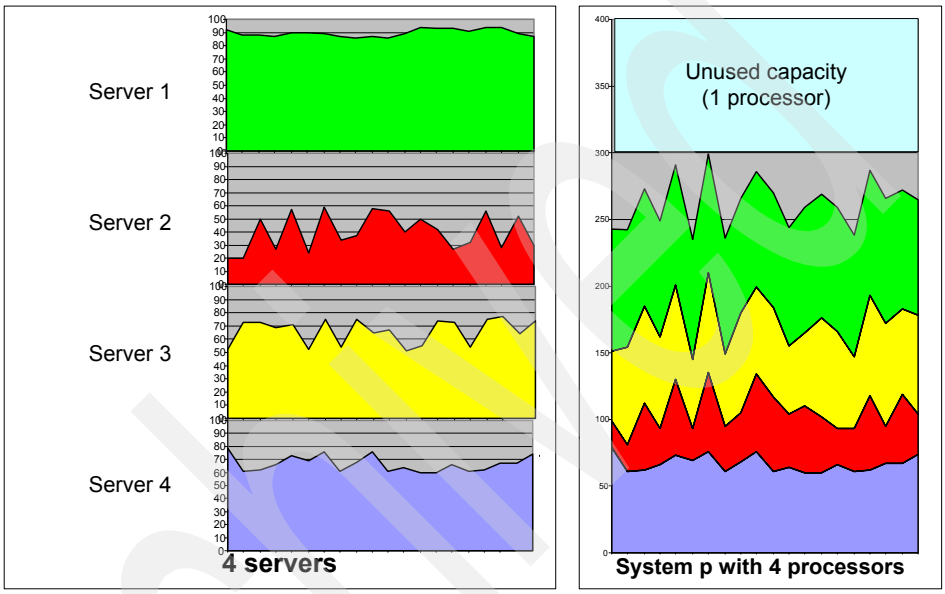


Figure 4-17 Server consolidation made available through micro-partitioning

Setting a partition to share its unused dedicated processor capacity is performed through the partition profile settings. To change this setting, edit the profile of the partition to be changed, navigate to the Processors tab, and select the desired check boxes for Sharing Processors. Figure 4-18 shows these settings. The processor can be shared when it is active, inactive, or both.

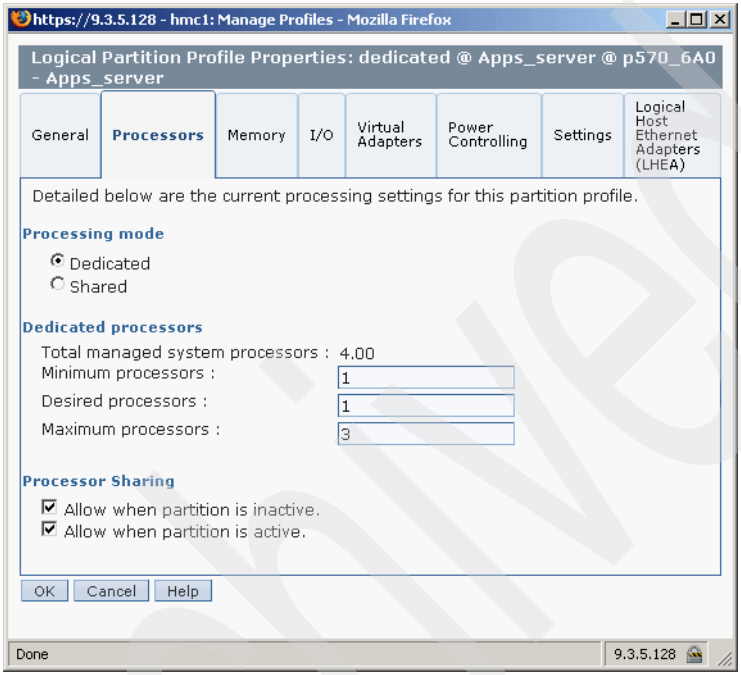


Figure 4-18 Setting processor sharing for dedicated processors

To activate the Virtual I/O Server (VIOS), the PowerVM feature and a logical partition with enough resources to share with other partitions must be available on your System p.

Due to the characteristics of the virtualization features of System p servers, the operating system and application do not realize that they are running in either a micro-partition or a virtualized I/O environment. This allows applications to run unmodified in partitions that take advantage of both features. Additionally, because VIOS partitions handle the translation of the virtual adapters I/O operation to the physical adapter, you need to make sure that this partition is sized properly to handle I/O requirements in all partitions.

The IBM Systems Hardware Information Center provides a starting point for the detailed planning calculations needed for CPU and memory planning:

http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphb1/iphb1_vios_planning.htm

System workload, such as network and disk usage, has sporadic activity because there are bursts of network traffic and disk activity when actions are performed. For example, when a client accesses a Web site and submits changes, some network use is generated with a burst of disk activity to update a back-end database. For this reason, making full use of the micro-partitioning and uncapped CPU features within the VIOS makes the most sense. A guideline is to use 1 GB of memory as a starting point for the VIOS, and scale down or up from there. The CPU can take a bit more thought, but the only way to guarantee accuracy is to run the system under real workloads and then monitor performance for tuning.

If you plan dual VIOS, a bit of planning can make sure that you can size two smaller Virtual I/O Servers that support half of the virtual I/O clients each. Additional capacity to allow for virtual I/O resilience can be provided through the uncapping facility of the servers. With redundant VIOS environment, client partitions can access a physical adapter in two different VIOS partitions, through the definition of multiple virtual adapters. This environment for client partitions can take advantage of high availability features such as multi-path I/O software or link aggregation technologies, such as Etherchannel, and the whole partition can continue to operate properly even in the case of a fault of the VIOS level or even in the external network or storage devices connected to the server.

The strategic planning of using dual Virtual I/O Servers also allows you to do maintenance on one Virtual I/O Server while the other one continues to provide production services.

4.4.5 Simplified management

SOA infrastructure complexity is based on the loose coupling and distributed nature of the architecture. It requires a robust set of management functions that allows quick deployment and operation of the components. It also allows for a more structured way of monitoring the application. The manageability of System p has been described in 4.2.4, “System p management features” on page 119.

The physical maintenance of systems depends on the number of servers to manage. Virtualization drives the numbers down to servers with virtual systems running on them. The management of a virtualized environment leads to an efficient provisioning of resources based on the business demands.

As the management of these resources is made easy, the associated effort of all related tasks, like provisioning and server creation, is being limited to a minimum. This allows administrators to focus on business critical management tasks.

Management across the whole enterprise gets much easier with System p. Partition and application mobility allows an easy way to move systems based on scaling or organizational requirements.

Hardware management

Hardware management is performed by the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM). The HMC provides full management functions across multiple System p servers from a single console. IVM is hosted on a partition on a single System p server. IVM provides a subset of HMC functions and a single point of control for small system virtualization.

As discussed in “Deployment of virtual servers” on page 151, deployment and provisioning is managed by the system plans. New virtual servers (partitions) can be deployed using predefined templates. The use of templates means that a Network Installation Manager (NIM) server must be present.

Workload management and partitioning

Enterprise Workload Manager provides methods to capture particular information gathered directly from specific applications. This is not System p specific, but can support decisions in System p management effectively.

Through cooperation between the participating application server processes and operating system images, data is collected, aggregated, displayed, and logged. In addition to these basic functions, a new level of intelligence is added to provide recommendations for workload routing decisions made by routers and middleware products, which can help the environment avoid known weaknesses of the participating operating system platforms. Cooperative support is provided with edge servers to correlate network quality-of-service adjustments with workload routing, balancing, and management decisions made by the attached server farm. Security features, scalability, robustness, and ease-of-use functions needed for production are also provided in this release.

Enterprise Workload Manager is also a requirement for using the workload partitioning. Workload Partition allows management of many applications under one operation system by providing an abstraction of the application towards the operating system. More about workload partition can be found in Appendix B, “Workload partitions benefits” on page 227.

Monitoring

For determined and successful systems management, the feedback of the IT applications and systems hosted on a hardware component must be collected and analyzed. There are several options that can be used to gather information about the system and applications.

IBM Director

IBM Director includes a set of tasks that can be used to gather direct feedback from the application or the system that is supporting the SOA operational systems layer.

In IBM Director, some functionality is available without an agent (Level-0 systems), working only with the underlying operating system. Systems with IBM Director Core Services installed (Level-1 systems) provide a basic set of tasks. Systems with the full IBM Director Agent installed (Level-2 systems) provide the full suite of tasks included with the base IBM Director offering.

In addition to the three agent levels described above, there are extensions (also called subagents or plug-ins) that are available from IBM and third parties that further extend the capabilities of IBM Director. Some of these extensions are provided on the base IBM Director installation CD and are simply selected during installation.

IBM Director has a special extension for System p in the form of the Virtualization Manager plug-in. The Virtualization Manager allows HMC integration.

The following features provide the ability to gather information about the application or system status and are supportive of SOA maintenance and governance operations.

- SNMP browser

  With the help of the SNMP browser, applications that offer an SNMP interface can be monitored directly inside of IBM Director. Based on such information, infrastructure changes can consider application scope rather than just focusing on information from the system directly, such as 90% CPU usage.

  Basically, this can give the SOA a context for systems management by considering the application level usage of resources and doing operations based on context relevant information. This will also lead to a much more targeted approach of scaling and managing systems and resources.

- External application launch

  This feature allows you to incorporate external applications to be launched inside of IBM Director. It can be used additionally to gather specific information from applications or execute operations on services, middleware, and other related components.

- Resource monitors

  Resource monitors basically monitor the available resources on the system and are not able to gather any application scope specific data. But their capability to monitor processes and TCP/IP specific resource usage might be helpful in conjunction with other data.

For more information about IBM Director, refer to *Implementing IBM Director 5.20*, SG24-6188.

Tivoli Monitoring and Tivoli Composite Application Manager

The IBM Tivoli Composite Application Manager family complements the monitoring up to the Web services layer of the IT infrastructure. More information about this product family can be found in *IBM Tivoli Composite Application Manager Family Installation, Configuration, and Basic Usage*, SG24-7151.

5

SOA implementation

This chapter discusses the hardware and software implementation of our SOA scenario. This discussion only represents a specific configuration for our scenario. There can be many more scenarios and options for implementing SOA that are not covered here, but this discussion can serve as a starting point by providing examples of some common tasks. We discuss the following topics:

- 5.1, “Implementation environment” on page 168
- 5.2, “Hardware configuration” on page 169
- 5.3, “Software environment” on page 189

5.1 Implementation environment

We implement our SOA sample environment on two sets of System p partitions. The first set is running AIX and the other is running Linux. The overall machine configuration is shown in Figure 5-1.

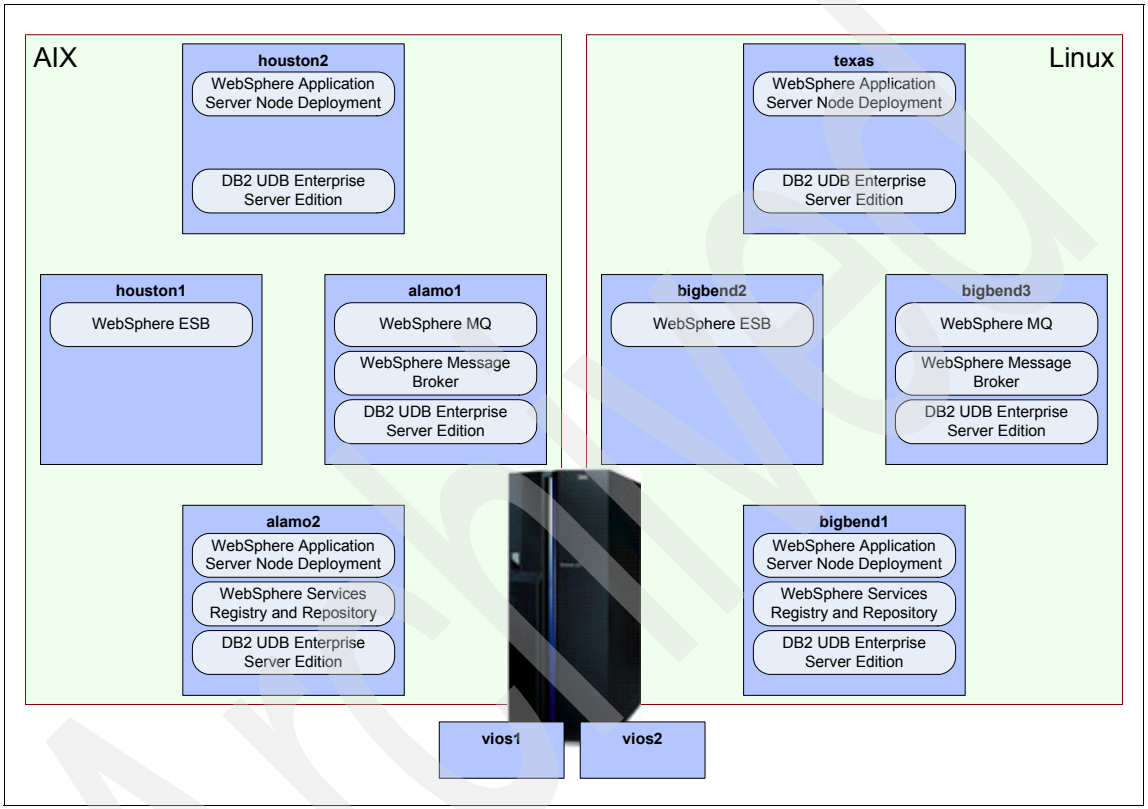


Figure 5-1 Implementation environment

We discuss more of the application environment in Chapter 6, “Operational environment” on page 197.

Table 5-1 on page 169 lists the hardware partitioning that we define for each partition.

Table 5-1 Hardware partitions

Partition	Processor	Memory	Disk
houston1			
houston2			
alamo1			
alamo2			
texas			
bigbend1			
bigbend2			
bigbend3			
vios1			
vios2			

Table 5-2 lists the software levels that we use.

Table 5-2 Software level

Product	Version
WebSphere Application Server Network Deployment	6.1.0.13
WebSphere Enterprise Server Bus	6.1.0.0
WebSphere Message Broker	6.0.0.3
UDB DB2 Enterprise Server Edition	9.1.0.0
WebSphere MQ	6.0.0.0
WebSphere Service Registry and Repository	6.1.0.0

5.2 Hardware configuration

This section discusses the configuration of our System p server. The discussion includes:

- 5.2.1, “System plan” on page 170
- 5.2.2, “Virtual servers” on page 172
- 5.2.3, “System plans in Virtualization Manager” on page 175
- 5.2.4, “Virtual I/O Server partition” on page 178

A system plan is a specification of the hardware and the logical partitions that are contained in one or more systems. This section explains how to validate the system plan, how to import the system plan (using both the HMC and the IVM), and how to deploy a system plan (using both the HMC and the IVM).

5.2.1 System plan

When validating the hardware on the managed system, the HMC compares the following information from the system plan with the hardware that is available on the managed system:

- Amount of processor and memory
- Physical I/O adapter placement

The hardware that is described in the system plan passes validation if it matches the hardware that is specified by the managed system. The hardware on the managed system can contain resources in addition to those specified in the system plan and still pass validation. However, the hardware on the managed system must at least match the hardware that is specified in the system plan.

For example, a system plan specifies a server with two processors, 8 GB of memory, and a specific placement of physical I/O adapters within the system unit. A server that contains two processors, 16 GB of memory, a matching placement of physical I/O adapters within the system unit, and an expansion unit with additional physical I/O adapters allows the system to pass validation. A server that contains 4 GB of memory causes the system to fail validation. A system plan also fails validation if the system plan specifies one type of physical I/O adapter in a slot, but the actual system unit has a different type of physical I/O adapter in that slot. (However, if the system plan specifies an empty slot, validation allows any type of physical I/O adapter to be in that slot on the actual system.)

The HMC does not validate the disk drives that are attached to physical I/O adapters against the disk drives that are specified in the system plan. You must ensure that the disk drives that are installed in the managed system support the desired logical partition configuration. Embedded devices pass hardware validation automatically because they are embedded into the system and cannot be removed.

When validating an existing logical partition, the HMC validates the following items for that logical partition. Validation fails for the existing logical partition if any step fails. Any existing partition found on the managed system must appear in the system plan and must match the system plan as it appears in the managed system.

Creating a system template based on an existing virtual server

Before creating a system template, ensure that the following requirements are met:

- An operating system is installed on the source virtual server.
- IBM Director has discovered and has access to the source virtual server.
- The source virtual server is powered on and has a valid IP address.
- If you are using an image repository that is on the NIM server, ensure that IBM Director has discovered and granted access to the managed system running the NIM server.

Creating a system template from an existing image file

Since the image repository is on the NIM server, ensure that IBM Director has discovered and has access to the managed system running the NIM server. To check this situation, simply look at the HMC view in the IBM Director Console. HMCs with access granted to IBM Director will not have a lock icon in the view. To request access to an HMC, right-click the HMC with the locked icon and select **Request Access...** Refer to Figure 5-2 to see the IBM Director HMC view with both locked and unlocked HMCs and how to request access.

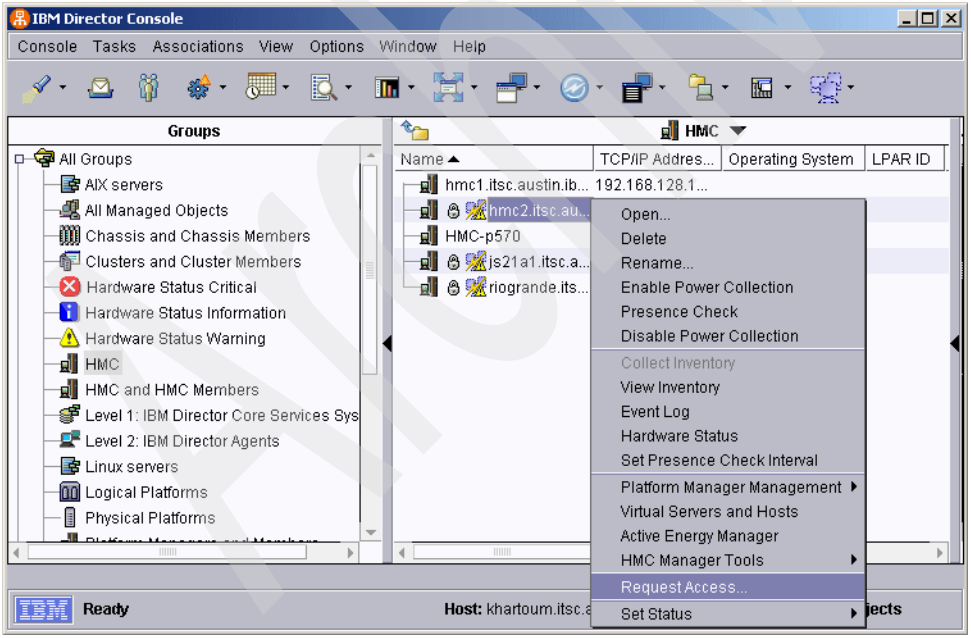


Figure 5-2 IBM Director HMC view showing access rights

Image repository

An image repository is part of a local or shared file system that is used to store system images. When you create a system template, you can specify the location of an image repository that holds the image you want to use for the system template. Also, you can specify the location of an image repository where you want to store the newly created image. When the image repository is on a NIM server, the system image is in the format of a mksysb. The managed system running the NIM Server requires the IBM Director Agent and the Virtualization Manager Agent for AIX NIM Server.

It is possible to create a system template for System p systems by pointing to a system image that is stored in an image repository on an AIX NIM server.

5.2.2 Virtual servers

The tasks discussed here are:

- “Creating a virtual server” on page 172
- “Using system templates” on page 173
- “Creating a system template” on page 173
- “Deploying a system template” on page 174
- “Editing a system template” on page 175

Creating a virtual server

You can use the Create Virtual Server wizard in IBM Systems Director Virtualization Manager to create virtual servers on your host systems. The wizard guides you through the virtual server creation process. The wizard prompts you to provide information such as the name, processing units, memory, and storage to allocate to the virtual server. The information it requests is specific to the virtualization environment on which the virtual server is being created.

To create a virtual server, complete the following steps:

1. From IBM Director Console, in the Group Contents pane, right-click the managed object for the host.
2. Click **Host Management** → **Create Virtual Server**.
3. Complete the instructions in the Create Virtual Server wizard.

After you create a virtual server on a system template that contains an image, you need to complete some additional steps to ensure that the virtual server was created and the image was installed on the virtual server. Creating a virtual server on System p systems based on a system template that contains an image involves the following underlying processes:

- Creating the virtual server
- Installing the system image on the new virtual server
- Verifying that the virtual server was created

To verify that the virtual server was created and the image was installed on the virtual server, complete the following steps:

1. After the Create Virtual Server wizard completes, wait five minutes. Then use Resource Navigator in the Virtualization Manager Web interface to locate the new virtual server.
2. If the virtual server is not displayed, try to discover it by clicking the context menu that appears to the right of the host system and selecting **Hardware and Software → Hardware Management → Discover Virtual Servers**.
3. If you are unable to discover the virtual server, verify whether an Event Log was created in IBM Director. In the Virtualization Manager Web interface, click the context menu that appears to the right of the host system and select **Health → Event Log**. If errors occurred during the process of creating a virtual server, a Task Failed event is displayed in the Event Log.
4. After the virtual server has been created and you can view it in the Virtualization Manager Web interface, monitor the image installation process to verify that the process has started by using the Hardware Management Console (HMC) or the Integrated Virtualization Manager (IVM).

Using system templates

Create, edit, and deploy system templates in your environment to create virtual servers. In some cases, you need to prepare your system for system template creation.

Creating a system template

Use the Create System Template wizard in IBM Systems Director Virtualization Manager to create system templates for creating virtual servers. The wizard guides you through the system template creation process. The wizard prompts you for the following information:

- Name of the system template.
- Whether you want to use an image to create the system template.

- Type of image you want to use, such as a new image, existing image, or system plan.
- Location of new or existing image or system plan.
- Storage location for new image. The storage location must have an image repository that supports the operating system on the image you specified for the system template.
- Allocated resources, such as processing units, memory, and storage to be used for the virtual server.
- Optional customized settings that affect the operating system and TCP/IP settings.

To create a system template, complete the following steps:

1. In the Virtualization Manager Web interface, expand **Templates and Deployment**.
2. Click **Create System Template**.
3. Follow the instructions in the Create System Template wizard.

Deploying a system template

Deploy a system template to create a new virtual server based on the selected system template. For most system template functions that involve images, you need to have an active license for IBM Systems Director Virtual Image Management. To deploy a system template, complete the following steps:

1. In the IBM Systems Director Virtualization Manager Web interface, expand **Templates and Deployment**.
2. Click **System Templates**.
3. Click the context menu that appears to the right of the system template you want to deploy and select **Deploy System Template**.
4. Complete the instructions in the Deploy System Template wizard.

When deploying a template that represents a system plan, the Deploy System Template wizard enables you to launch a Hardware Management Console (HMC) or the IBM Integrated Virtualization Manager (IVM) to deploy a system plan from either of those platform managers. If the selected system plan represents multiple virtual servers, the Deploy System Template wizard will create all the virtual servers for the host instead of just one. When deploying a template that points to a master image for POWER, a Virtual I/O Server must exist on the host system where the template is deployed. The Virtual I/O Server is required to create the virtual disk that holds the image for the new virtual server.

Editing a system template

Edit the settings of a system template by using the IBM Systems Director Virtualization Manager Web interface. If your system template points to a system plan for creating virtual servers on POWER systems, this task is not enabled in the Web interface. To edit a system template, complete the following steps:

1. In the Virtualization Manager Web interface, expand **Templates and Deployment**.
2. Click **System Templates**.
3. Click the context menu that appears to the right of the system template you want to edit and select **Edit**.
4. Edit the settings of the system template.

5.2.3 System plans in Virtualization Manager

You can generate, import, and export system plans using Virtualization Manager. The following describes these tasks in detail:

- "Using system plans on the HMC or IVM" on page 175
- "Generating a system layout (system plan)" on page 175
- "Importing a system plan" on page 177
- "Exporting a system plan" on page 178

Using system plans on the HMC or IVM

Generate system layouts and import and export system plans between the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) and other systems. Figure 5-3 shows the partitions summary page of a system plan view based on an example configuration of a POWER6 570 server. In it you can see the processor and memory settings of the profiles configured for the system. Additional information may be viewed by clicking the partition names in the navigator pane.

Generating a system layout (system plan)

Virtualization manager allows you to generate a system plan for System p servers. System plans are referred to as system layouts within Virtualization Manager. You can use the system plan to quickly configure newly purchased managed systems that are similar to the one where the plan was created. After generating a new system plan, you can view the system plan in the System Plan Viewer and use it to create a system template for creating new virtual servers.

To generate a system layout (system plan) from Virtualization Manager, complete the following steps:

1. In the IBM Systems Director Virtualization Manager Web interface, expand **Hardware and Software** → **Virtual Servers and Hosts**.
2. Click the context menu that appears to the right of the system from which you want to create a system layout and select **Generate System Layout**.
3. Optional: Select **Import Generated System Plan** in the template list if you want Virtualization Manager to create a system template that you can view in System Templates.
4. In the Name field, type a name for the new system plan.
5. In the Description field, type a description for the new system plan.
6. Click **Generate Now**. The system plan might take several minutes to several hours to generate. Virtualization Manager monitors the completion of these processes for approximately two minutes.
   - If the system plan process completes quickly, a message is displayed that indicates whether the system plan was created successfully. If the system plan is created successfully, you can click **System Plan Viewer** to open the System Plan Viewer and view the new system plan. If the system plan is not created successfully, go to the IBM Director Event Log to see error messages from the Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM).
   - If the process runs longer than approximately two minutes, a message is displayed that indicates the process is still in progress. After the request finishes, an event will be added to the IBM Director Event Log for the host resource and you can view the system plan in the System Plan Viewer. To check the Event Log from the Virtualization Manager Web interface, click the context menu that appears to the right of the host system and select **Health** → **Event Log**. Each system plan is contained within a system plan file with a file suffix of .sysplan.

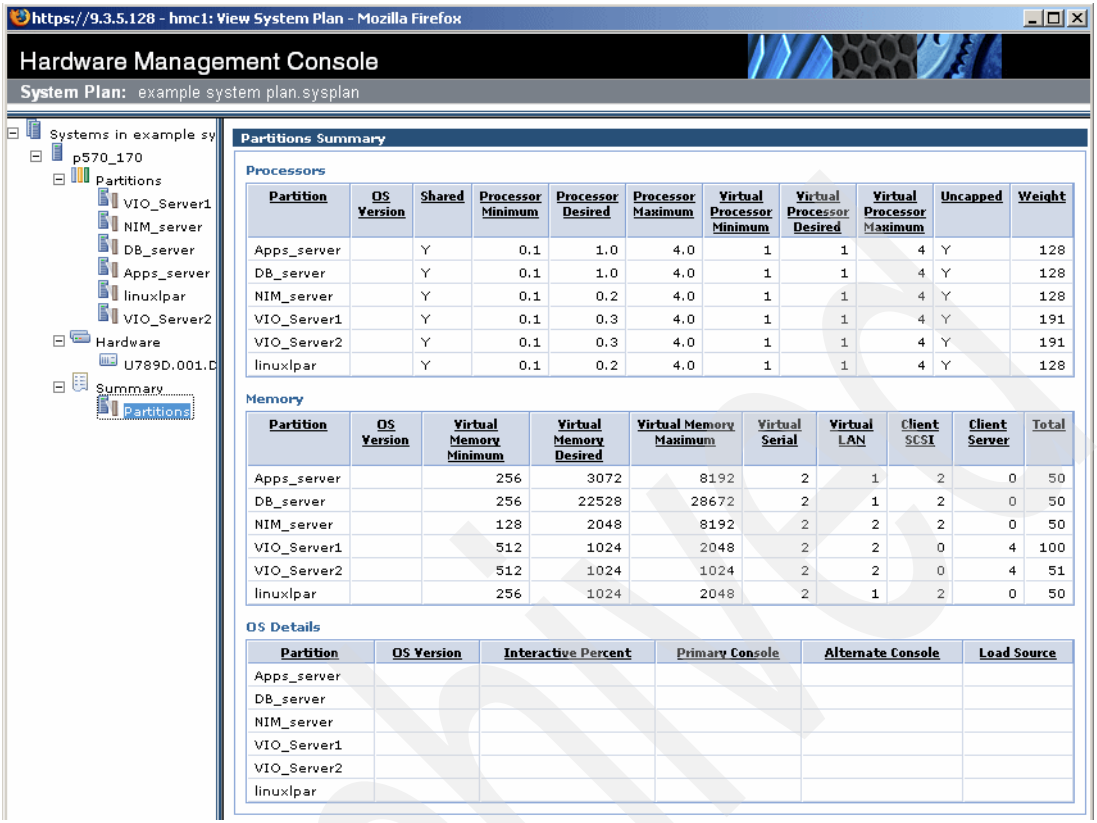


Figure 5-3 Partitions Summary view of system plan

Importing a system plan

You can import a system plan into an Hardware Management Console (HMC) or Integrated Virtualization Manager (IVM) so you can later deploy the system plan to managed systems that are managed by the HMC or IVM. You can import a system plan into an HMC from media, a remote FTP site, or the computer from which you remotely access the HMC. To import a system plan into an IVM, you may import from the computer from which you remotely access the IVM. To import a system plan, complete the following steps:

1. In the IBM Systems Director Virtualization Manager Web interface, expand **Hardware and Software** → **Virtual Servers and Hosts**.
2. Click the context menu that appears to the right of the HMC or IVM into which you want to import a system plan and select **Import System Plan**.

The interface enables you to launch an HMC or IVM and provides instructions for completing the task.

Exporting a system plan

You can export a system plan from an Hardware Management Console (HMC) or an Integrated Virtualization Manager (IVM) to another system. For example, you can export the system plan to your desktop workstation to enable you to run the IBM System Planning Tool (SPT) against it. You can export a system plan from an HMC to media, a remote FTP site, or the computer from which you remotely access the HMC. From an IVM, you can export a system plan to the computer from which you remotely access the IVM. To export a system plan from an HMC or IVM to another system, complete the following steps:

1. In the IBM Systems Director Virtualization Manager Web interface, expand **Hardware and Software** → **Virtual Servers and Hosts**.
2. Click the context menu that appears to the right of the HMC or IVM from which you want to export a system plan and select **Export System Plan**. The interface enables you to launch an HMC or IVM and provides instructions for completing the task.

5.2.4 Virtual I/O Server partition

The following section describes how to use the Create LPAR Wizard to create a Virtual I/O partition. The steps are as follows:

1. Start Create LPAR Wizard
2. Enter Partition Information
3. Enter Partition Profile name
4. Select processor type
5. Enter Processor Settings
6. Enter Memory settings
7. Select Physical I/O adapters
8. Create Virtual Adapters
9. Create virtual SCSI adapter
10. Create virtual Ethernet adapter
11. Select optional settings
12. Review settings and create partition

Start Create LPAR Wizard

From the HMC Systems Management Servers view, select the server you want to create the partition on, and then select **Configuration** → **Create Logical Partition** → **VIO Server**. The Create Lpar Wizard appears. See Figure 5-4 for a screen capture of the first page of the Create Lpar Wizard.

Enter Partition Information

On the Create Partition page of the Create Lpar Wizard, enter a Partition ID and Partition name. It is best to use the default Partition ID. If this is a new system, then use 0 and 1 for the partition IDs for your Virtual I/O Servers. This will make administration simpler. Click **Next** when done. Figure 5-4 shows a Partition ID of 6 and a Partition name of vio1.

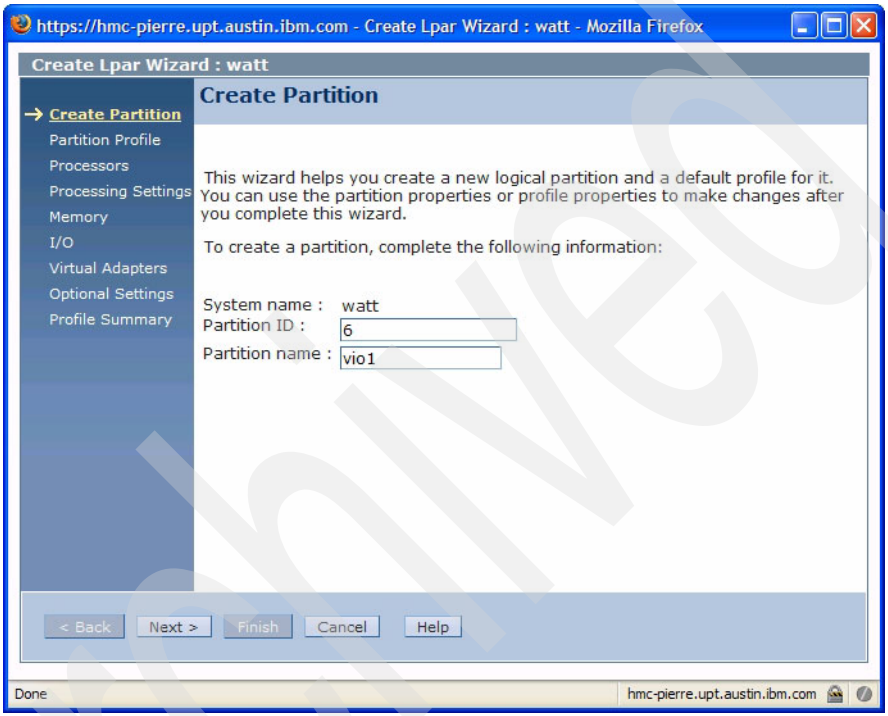


Figure 5-4 Create Lpar Wizard: Create Partition

Enter Partition Profile name

The next window shows the Partition Profile page of the Create Lpar Wizard. Enter the profile name, as shown in Figure 5-5.

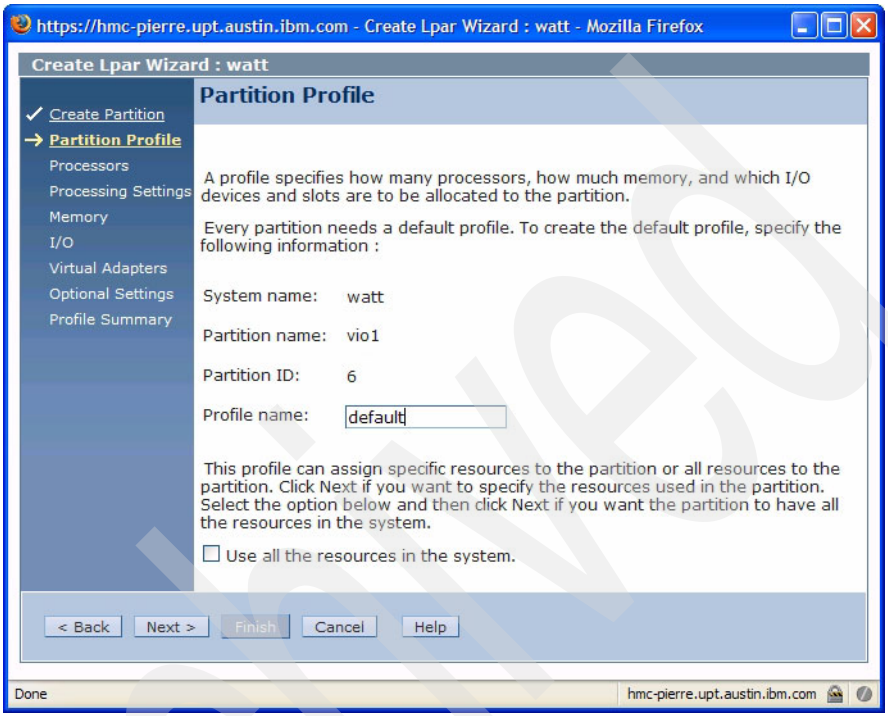


Figure 5-5 Create Lpar Wizard: Partition Profile

Select processor type

In the Processors window of the Create Lpar Wizard, select either **Shared** or **Dedicated**. Whether you choose shared or dedicated will be determined by your capacity requirements. For information about determining your capacity requirements, refer to *Virtualization and Clustering Best Practices Using IBM System p Servers*, SG24-7349. See Figure 5-6 for a screen capture of the Processors page of the Create Lpar Wizard. Click **Next** when done.

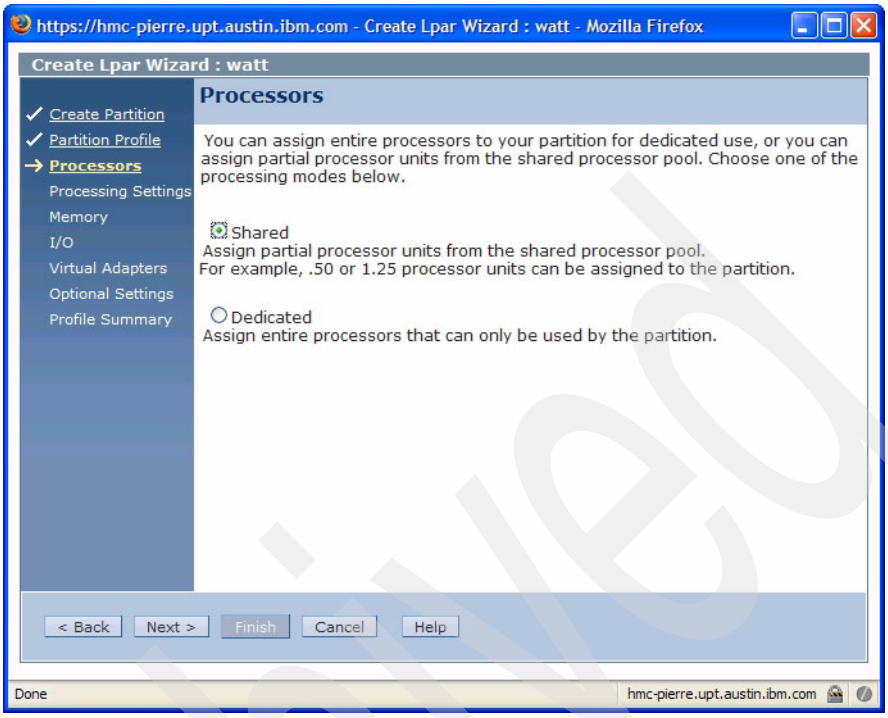


Figure 5-6 Create Lpar Wizard: Processors

Enter Processor Settings

Next, select the Processing Settings for your Virtual I/O Server. Set the minimum, desired, and maximum number of processing units. If you chose Shared Processor in the previous step, you will be able to set fractions of a processor as fine as one one-hundredth of a processor. Additionally, you may set the minimum, desired, and maximum virtual processor units. A virtual processor is a representation of a physical processor to the operating system that uses the shared processor pool. To understand how virtual processors work, refer to 4.2.1, “Virtualization on System p” on page 104. Finally, again for shared processors only, you may select whether the partition CPU capacity is uncapped. If **Uncapped** is selected, then enter the Weight. The Weight is a number in the range from 0 through 255 that you set for each uncapped partition in the shared processor pool. Using the HMC, you can choose from any of the 256 possible uncapped weight values. The Integrated Virtualization Manager and the Virtual Partition Manager limit you to only one of several different uncapped weight values. By setting the uncapped weight (255 being the highest weight), any available unused capacity is distributed to contending logical partitions in proportion to the established value of the uncapped weight. The default uncapped weight value is 128.

For example, logical partition 2 has an uncapped weight of 100, and logical partition 3 has an uncapped weight of 200. If logical partitions 2 and 3 both require additional processing capacity, logical partition 3 would receive two additional processing units for every additional processing unit that logical partition 2 receives. See Figure 5-7 for the Processing Settings page of the Create Lpar Wizard.

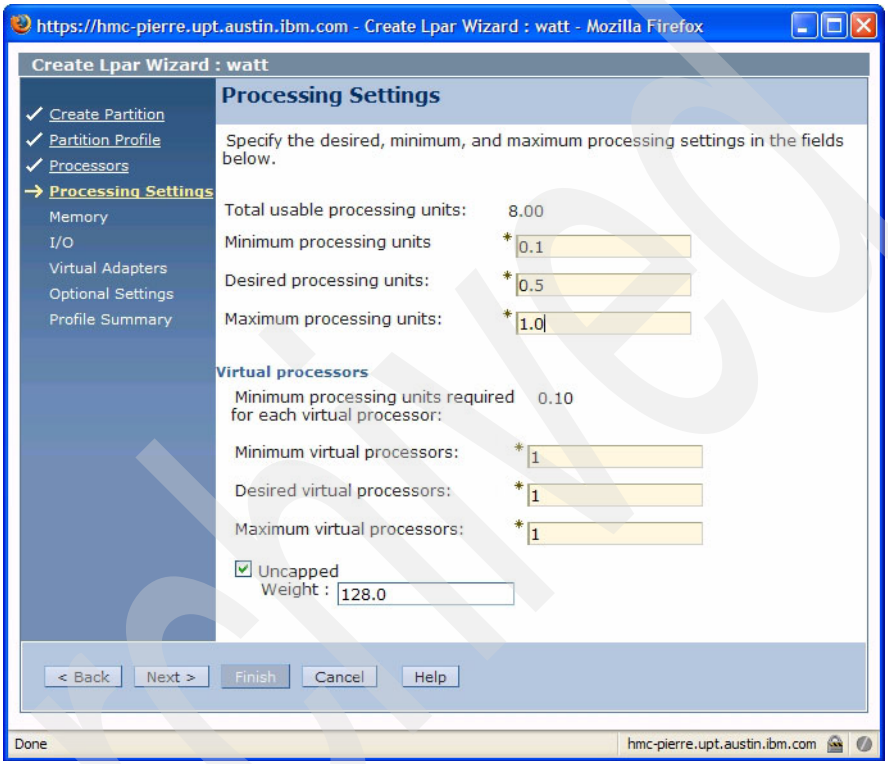


Figure 5-7 Create Lpar Wizard: Processing Settings

Enter Memory settings

The next page in the Create Lpar Wizard is Memory settings. Set the minimum, desired, and maximum amounts desired for your logical partition. See Figure 5-8 for an example of this page. The minimum, desired, and maximum memory shown in the figure are 256 MB, 512 MB, and 1 GB, respectively. Note that the current available memory is displayed for your reference. The amount of memory required for a Virtual I/O Server depends on the workload that will be served through the VIO Server, which usually depends solely on the amount of Ethernet traffic. To determine the amount of memory your Virtual I/O Server will require, refer to *Virtualization and Clustering Best Practices Using IBM System p Servers*, SG24-7349.

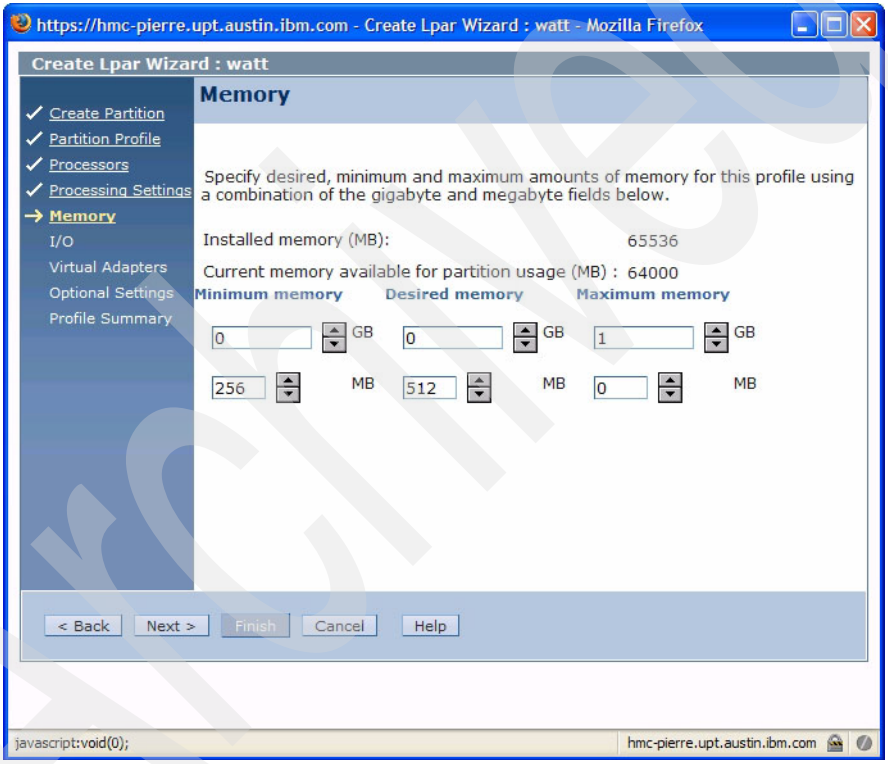


Figure 5-8 Create Lpar Wizard: Memory

Select Physical I/O adapters

The next window in the Create Lpar Wizard is I/O. Adding physical I/O totally depends on planning and server resources. Select the physical adapters that are necessary for connecting the Virtual I/O Server's client logical partitions to external physical networks or storage. Generally, you want physical adapters assigned to Virtual I/O Servers to be added as Required to prevent them from being accidentally dynamically removed through a DLPAR operation. See Figure 5-9 for an example of this window.

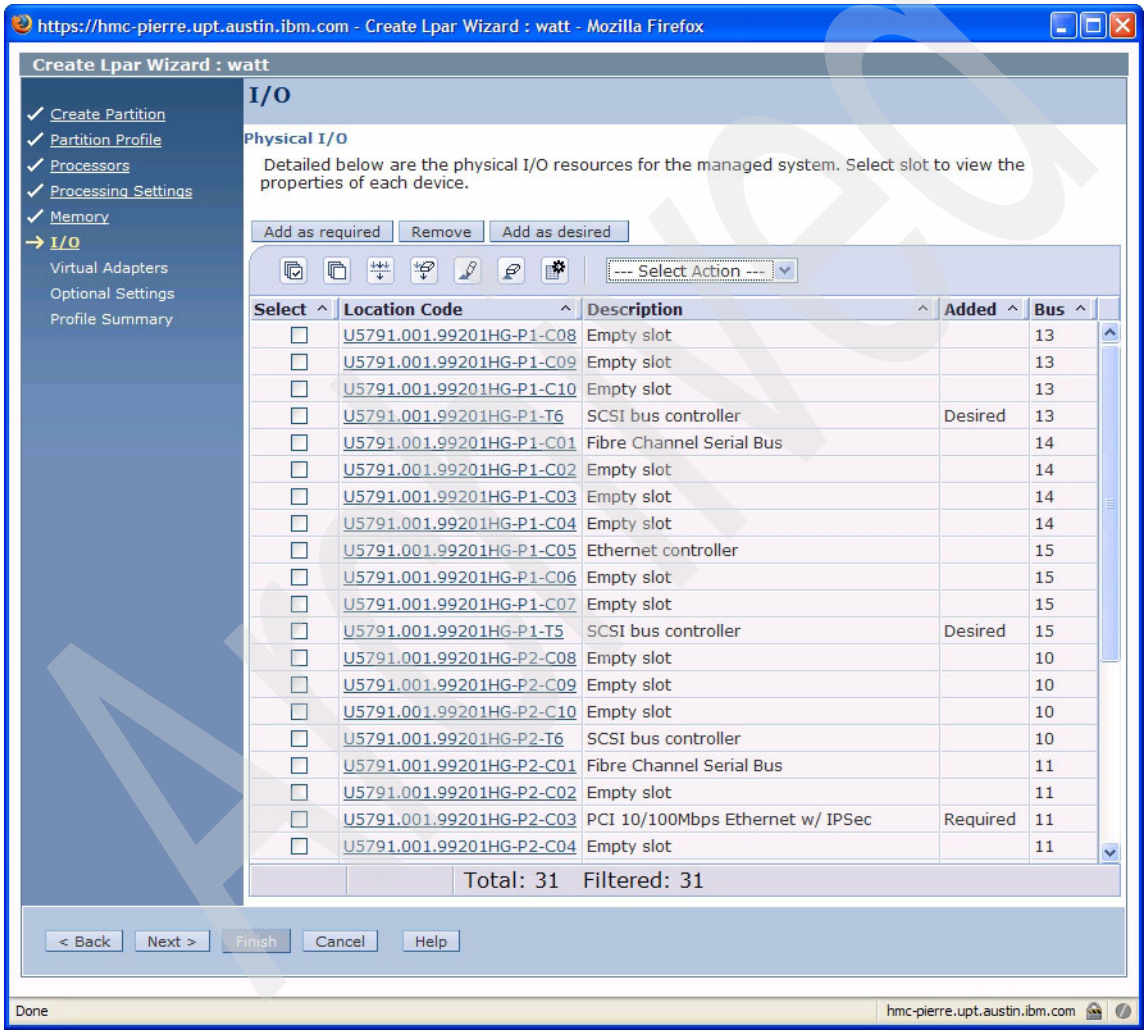


Figure 5-9 Create Lpar Wizard: Physical I/O

Create Virtual Adapters

The next page in the Create Lpar Wizard is Virtual Adapters. There are three types of possible Virtual Adapters: Serial, SCSI, and Ethernet. The virtual Server Serial adapters are created automatically. You will need to create virtual SCSI adapters if you are using the VIO Server to virtualize your storage. Likewise, you will need to create a virtual Ethernet adapter if you are using the VIO Server to virtualize your networking. The following sections refer to Figure 5-10, which shows the Virtual Adapters page of the Create Lpar Wizard.

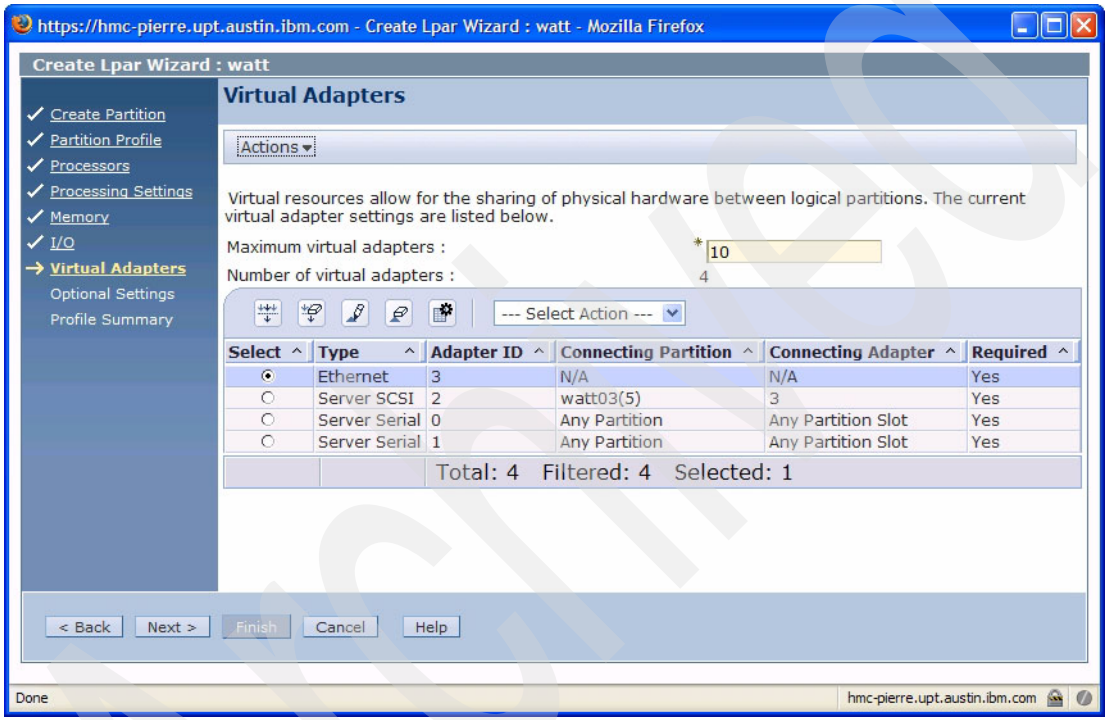


Figure 5-10 Create Lpar Wizard: Virtual Adapters

Create virtual SCSI adapter

To create a virtual SCSI adapter, select **Actions** → **Create** → **SCSI Adapter**. The Create Virtual SCSI Adapter dialog box appears. Set the adapter slot number you desire (usually the default is best). The slot number is used to identify the virtual adapter within the logical partition. The combination of the slot number and the logical partition ID uniquely identifies this slot within the managed system. Note that the slot number does not refer to any physical hardware location on the system. Use a system that makes most sense to you but remember that you may only use slot numbers from 2 up to the maximum number of virtual adapters.

Additionally, you may not use a slot number that is used for another virtual adapter on the same partition. Next, select whether the adapter is required for partition activation. Usually, this will be true except for rare usage scenarios. If you are unsure at this moment which client partition will connect to this virtual SCSI adapter, or if you have not created the client partitions yet, select **Any client partition can connect**. To prevent confusion and errors, and for a better structured environment, you will want to select a specific client partition for this particular adapter. Usually, you will want to select **Any client partition can connect** only when the storage being served up needs to be shared. If you select **Only selected client partition can connect**, you will need to select first the client partition, and second, but more important, the client adapter ID. If the client partition is created after the VIO Server, you will need to know this client adapter number. In HMC V7 and above, this number can be viewed through a dialog box during client partition profile creation. If the client partition has already been created, then you will need to open the client partition profile to view the slot number. Then you will use this same number here. See Figure 5-11 for the CreateVirtual SCSI Adapter dialog box showing some sample settings.

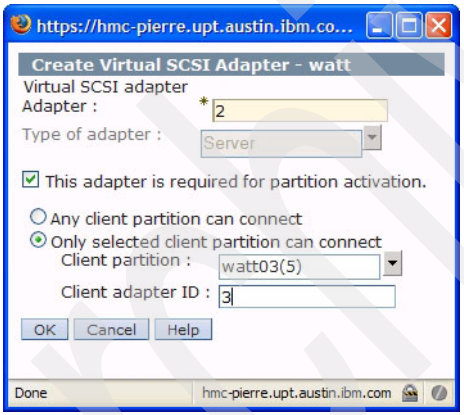


Figure 5-11 Create Lpar Wizard: Create Virtual SCSI Adapter

Create virtual Ethernet adapter

To create a virtual Ethernet adapter, from the Virtual Adapters page, select **Actions** → **Create** → **Ethernet adapter**. Enter the Adapter ID and VLAN ID. The virtual Ethernet Adapter ID is similar in function to the SCSI adapter ID. It uniquely identifies the adapter on the managed system. Usually, it is best to use the default unless you have developed a numbering system. Remember that it must be unique to the partition.

Next, enter the VLAN ID you wish to use. All virtual Ethernet adapters in a partition are associated with a Virtual Network (VLAN). A VLAN allows a physical network to be divided administratively into separate logical networks. In effect, these logical networks operate as though they are physically independent of each other. Your VLAN configuration will depend upon your network requirements. To get a better understanding of VLANs, refer to *Virtualization and Clustering Best Practices Using IBM System p Servers*, SG24-7349.

If your planning requires it, you may optionally select a Virtual Switch (VSwitch). VSwitches allow you to create multiple VLAN adapters in a partition and allow these adapters to route through different physical adapters to the external network. Each VSwitch is associated with a unique set of VLAN networks. VLANs are associated with one and only one VSwitch. Therefore, VLAN IDs can be reused from one VSwitch to another. Even though two VLANs may have the same VLAN ID, they represent different VLANs if they are associated with different VSwitches. The combination of VSwitch and VLAN ID identifies the VLAN uniquely. For example, VLAN 1 in VSwitch1 is not the same VLAN as VLAN 1 in VSwitch2.

If you do not want this adapter to be removed through a DLPAR operation, then select **This adapter is required for partition activation**. Optional Settings allow you to select whether the adapter is an IEEE 802.1Q compatible adapter or whether the adapter communicates with an external network. The IEEE 802.1Q setting allows the partition to communicate with multiple VLANs. If **Access external network** is selected, then the adapter is effectively the trunk adapter that links the VLAN(s) to the external network. Selecting this option will not automatically connect the virtual LAN to an external network. You must also configure the operating system of the logical partition to forward information to the external network. Also note that a VLAN may only have one active trunk adapter at a time, but redundant trunk adapters may be configured for fail-over capability. The trunk priority indicates the order in which this trunk adapter will be activated if you create redundant trunk adapters for virtual trunk adapter fail-over capability.

Refer to Figure 5-12 for the Virtual Ethernet Adapter settings.

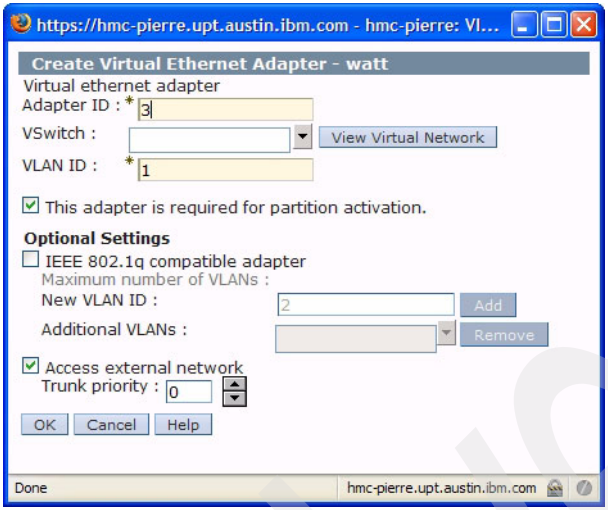


Figure 5-12 Create Lpar Wizard: Create Virtual Ethernet Adapter

Select optional settings

The next window in the Create Lpar Wizard is Optional Settings. Select the **Enable connection monitoring** option to enable frequent connection monitoring between the HMC and the logical partition. This option only controls whether the Service Focal Point (SFP) application automatically tests the connection and generates a serviceable event if the channel does not work.

Select **Enable redundant error path reporting** to enable the reporting of server common hardware errors from this logical partition. The service processor is the primary path. This option allows this logical partition to report server common hardware errors and partition hardware errors to the HMC. If this option is not selected, the logical partition reports only partition hardware errors to the HMC.

And finally, you may select whether the partition is automatically started when the managed system is booted, and the boot mode for the partition.

Review settings and create partition

The final page is the summary of the profile, shown in Figure 5-13. If any errors were made, you are able to go back and correct them before the managed system creates the profile. Click **Finish** to create the partition. If there are errors, the partition will not be created and you will be able to go back and correct the errors.

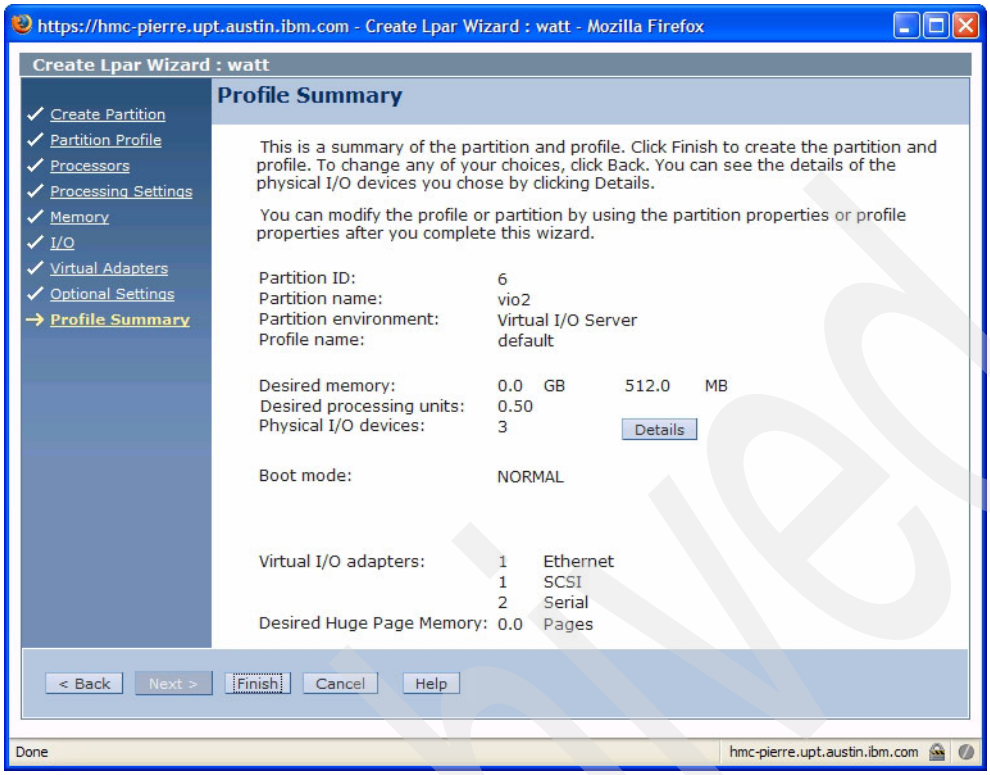


Figure 5-13 Create Lpar Wizard: Summary

5.3 Software environment

Companies in every industry are seeking to respond more quickly and effectively to changing customer and partner needs. The kind of business and information technology flexibility can be addressed by implementing a service-oriented architecture (SOA). SOA allows you to reduce the cost of development and increase the return of investment (ROI).

SOA implementation planning is critical, because it helps define key players in future business processes, the high impact services, and the corresponding solution architecture for realizing a complete SOA solution.

The SOA implementation plan helps define:

- Solution architecture
- Governance model
- SOA infrastructure readiness for successful realization of SOA solutions

A successful implementation of a SOA requires considerable resources, including expertise for designing the underlying IT infrastructure needed to support a SOA.

Based on the system diagram discussed in 5.1, “Implementation environment” on page 168, we discuss the installation tips for the following programs:

- 5.3.1, “WebSphere Application Server V6.1” on page 190
- 5.3.4, “WebSphere Message Broker V6.1” on page 194
- 5.3.2, “WebSphere MQ V6” on page 193
- 5.3.3, “DB2 Universal Database Enterprise Server Edition V9.1” on page 193
- 5.3.5, “WebSphere ESB V6.1” on page 195
- 5.3.6, “WebSphere Services Registry and Repository V6.1” on page 196

5.3.1 WebSphere Application Server V6.1

WebSphere Application Server Network Deployment is installed using the launchpad. Run `launchpad.sh` from a UNIX terminal. Select **WebSphere Application Server Network Deployment Installation**, as shown in Figure 5-14.

Figure 5-14 WebSphere Application Server Network Deployment launchpad

Follow the steps as described at the following Web page:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tins_install.html`

If you run the **`launchpad.sh`** command on IBM AIX Version 6.1, it might not bring up a launchpad console for installing IBM WebSphere Application Server Version 6.0 or Version 6.1. This is caused by shell security enhancements in AIX Version 6.1. There is a difference in the processing of the **`typeset +r LOGNAME`** command, so **`launchpad.sh`** fails to produce a window for installing WebSphere Application Server Version 6.0 or Version 6.1.

To use the launchpad to facilitate the installation of WebSphere Application Server Version 6.1, follow these instructions:

1. Extract the WebSphere Application Server installation image to a location on your AIX Version 6.1 system disk.

2. Install the bash shell on your AIX system if it is not already installed. You can download the bash RPM package from the Linux toolbox at:

   http://www.ibm.com/systems/p/os/aix/linux/toolbox/download.html

3. Download and unzip the launchpad_v2.0_fixpack18.zip attachment file to a directory on your AIX system from:

   http://www-1.ibm.com/support/docview.wss?uid=swg21288640&aid=1

4. Change to the launchpad_v2.0_fixpack18 subdirectory that contains the launchpad.sh file, and issue the following commands:

```
chmod +x *.sh
chmod +x launchpad/*.sh
chmod +x launchpad/content/*
chmod +x /usr/sbin/bootinfo
```

   Note: The last **chmod** command is only necessary if a non root user is installing WebSphere Application Server Version 6.1.

5. Copy all the files in the temporary directory over the top of your existing launchpad installation:

```
cp -r * your_directory/disk1/
```

6. Go to the WebSphere Application Server Version 6.1 installation-image directory and run the **./launchpad.sh** command.

To install WebSphere Application Server Fix Packs, we must install the Update Installer first. This is not packaged as part of the Fix Pack download and needs to be acquired separately. To get the Update Installer needed to install a Fix Pack, do the following Instructions:

1. For WebSphere maintenance level 6.1.0.13, you must use the Update Installer V6.1.0.13 or later. The latest update installer can be downloaded from:

   http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg24012718

2. Extract the downloaded Update Installer package (ZIP file) to a temporary directory.
3. Navigate to the UpdateInstaller subdirectory that was created in the previous step.
4. Use the install wizard to install the Update Installer by running the **install.sh** file.
5. Now we can continue to install the Fix Pack 13 for WebSphere Application Server using the update installer.

5.3.2 WebSphere MQ V6

WebSphere Message Broker requires the installation of WebSphere MQ. It handles the communication between the configuration manager, the brokers (Linux and AIX), and the business applications. It is a required prerequisite for any of the WebSphere Message Broker runtime components. A separate WebSphere MQ Queue Manager is required for each broker that is created on a system. The Configuration Manager and the User Name Server also require a queue manager, but is possible to use the same WebSphere MQ Queue Manager as one of the brokers.

WebSphere Message Broker Toolkit can communicate with the Configuration Manager without WebSphere MQ being installed. WebSphere MQ is required to create a broker domain or the default Configuration Manager.

It is not intention of this book to demonstrate the full installation process of WebSphere MQ. If you need help installing it, follow the instructions found at:

http://www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=GC34647800

You can get the latest fixes for WebSphere MQ at:

http://www-1.ibm.com/support/docview.wss?rs=171&uid=swg21254675

5.3.3 DB2 Universal Database Enterprise Server Edition V9.1

DB2 Universal Database Enterprise Server Edition is supported as a repository for WebSphere Message Broker configuration data, WebSphere Services Registry and Repository, and our sample application.

When installing DB2, you have three options. They are:

- Compact installation
- Typical Installation
- Custom Installation

The compact installation is adequate for all the functionality that we require. However, we chose the custom installation to be able to pick and choose the installation options that we want.

For a complete guide to installing DB2, go to:

http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.uprun.doc/doc/t0008875.htm

5.3.4 WebSphere Message Broker V6.1

We perform a full installation of WebSphere Message Broker. This requires WebSphere MQ and DB2 Universal Database to be installed. We install all the runtimes components of WebSphere Message Broker.

When installing IBM Message Broker on Red Hat Enterprise Server V5.1 for 64-bit POWER PC architecture (ppc64), you may have some problems with the libraries. Some libraries used by IBM Message Broker are not compatible with the Red Hat Enterprise Server V5.1 ppc64 libraries.

You must install compat-libstdc++33-2.2.3-61.ppc64.rpm, because it is a prerequisite for IBM Message broker installation. The command to install it is:

```
rpm -ihv compat-libstdc++33-2.2.3-61.ppc64.rpm
```

The RPM installation creates libstdc++.so.5 for PPC64, but IBM Message Broker presents an error such as:

```
Error libstdc++.so.5: ELFCLASS64: 87568
```

We then back up libstdc++.so.5 from /usr/lib and /usr/lib64, as shown in Example 5-1.

Example 5-1 Back up libstdc++.so.5

```
cd /usr/lib
mv libstdc++.so.5 libstdc++.so.5.old
cd /usr/lib64
mv libstdc++.so.5 mv libstdc++.so.5.old
```

Force the installation of the 32-bit version of libstdc++ from the RPM package compat-libstdc++33-2.2.3-61.ppc.rpm with the following command:

```
rpm -ihv compat-libstdc++33-2.2.3-61.ppc.rpm -force
```

You now have 32-bit libraries of libstdc++.so.5 in /usr/lib that IBM Message Broker can use. You can restore this 32-bit version with IBM Message Broker or you can put it in a different directory and set the LD_LIBRARY_PATH environment variable to that directory.

You can find the RPM packages on the Red Hat 5 DVD or CD number 3.

There are some additional security requirements for installing WebSphere Message Broker:

- The Broker user ID password has to be eight characters or less. This is mandated by DB2 installation usage.

- The Broker user ID must be a member of the mqbrkrs, mqm, and db2grp1 groups.
- When using Message Broker Toolkit to connect to the Configuration Manager, you can get BIP0991E with BIP1711W specifying that the user cannot view the Configuration Manager Proxy.

  You must add an access control list (ACL) entry to the Configuration Manager for the user with the Message Brokers Toolkit local user ID. For example, if wbiadmin connects to the Message Brokers Toolkit on COMPUTERA, which belongs to DOMAINA, and is trying to connect to Configuration Manager CMA:

  – wbiadmin must exist as a user on the Configuration Manager system and belong to the mqm group.

  – wbiadmin could have one of the following ACL entries:

    • To create an ACL entry for wbiadmin to connect to with full Configuration Manager Proxy authority from any computer or domain, issue **mqsicreateaclentry CMA -u wbiadmin -a -x F -p**.

    • To create an ACL entry for wbiadmin to connect to with full Configuration Manager Proxy authority from COMPUTERA only, issue **mqsicreateaclentry CMA -u wbiadmin -m COMPUTERA -x F -p**.

    • To create an ACL entry for wbiadmin to connect to with full Configuration Manager Proxy authority from DOMAINA only, issue **mqsicreateaclentry CMA -u wbiadmin -m DOMAINA -x F -p**.

  If a domain name and computer name match, and an ACL entry specified the -m parameter, connection from either the domain or computer is acceptable, because they are not considered to be different users.

  ACLs are necessary for all connections to the Configuration Manager, so the user who issues a command such as **mqsideploy** from the command line must also have the appropriate ACL entries.

5.3.5 WebSphere ESB V6.1

A WebSphere ESB V6.1 installation can follow the same instructions as the installation for WebSphere Application Server Network Deployment, as shown in 5.3.1, “WebSphere Application Server V6.1” on page 190.

5.3.6 WebSphere Services Registry and Repository V6.1

WebSphere Service Registry and Repository comes with an installer that provides an out-of-the-box solution for numerous platforms. WebSphere Service Registry and Repository requires a database and WebSphere Application Server for its interface.

Before installing WebSphere Service Registry and Repository, administrators should consider a number of factors that affects how they install WebSphere Service Registry and Repository and what configuration tasks they need to undertake in order to make WebSphere Service Registry and Repository provide the desired capability in their service-oriented architecture (SOA).

There are two separate stages in the installation and deployment process:

- Installing

  This means running the shell file to place files on the file system. The default path is /opt/IBM/WebSphereServiceRegistry. The installer can be run in three different modes: GUI, Console, or Silent.

- Deploying

  This is performed using the **installall.sh** command from the install sub path of WebSphere Service Registry and Repository. It performs the following:

  – Creates the repository database.

  – Creates runtime resources in the database.

  – Adds the applications to the WebSphere Application Server runtime environment.

Note: If you use the option -skip-dbcreate with **installall.sh**, for example, to use a remote database, **installall.sh** would also skip creating the runtime resources. You have to create the resources manually.

if you need further information about installing WebSphere Service Registry and Repository, follow this link:

http://publib.boulder.ibm.com/infocenter/sr/v6r1/topic/com.ibm.sr.doc/twsr_installn17_2.html

6

Operational environment

In this chapter, we introduce an operational environment that best matches many typical solutions. Bringing this environment together with our trader example from 6.1, “ITSO trader example” on page 198, we show a typical structure and discuss the SOA and infrastructure considerations.

We focus mainly on features and solutions that illustrate how beneficial a synergy between the IT infrastructure and business is. We reach this ideal synergy by utilizing the capabilities of the infrastructure layer provided by System p.

This chapter discusses the following topics:

- 6.1, “ITSO trader example” on page 198
- 6.2, “Using SOA scenarios” on page 201
- 6.3, “The runtime environment” on page 207
- 6.4, “Application feedback and monitoring” on page 211

6.1 ITSO trader example

In this section, we explain a sample SOA solution based on a fictitious stock trading example. It is basically a Web-based, trading floor portfolio hosted by the ITSO trading company. It provides stock quotes, buying, and selling. This section discusses the business environment:

- 6.1.1, “Business view” on page 198
- 6.1.2, “Application view” on page 199

6.1.1 Business view

Here we introduce the company and its business view. The business context diagram shown in Figure 6-1 shows how the trading company is organized and relates to Business Partners and systems. We focus on the interactions that are relevant for just illustrating the application and the environment, even if there may be many more of them.

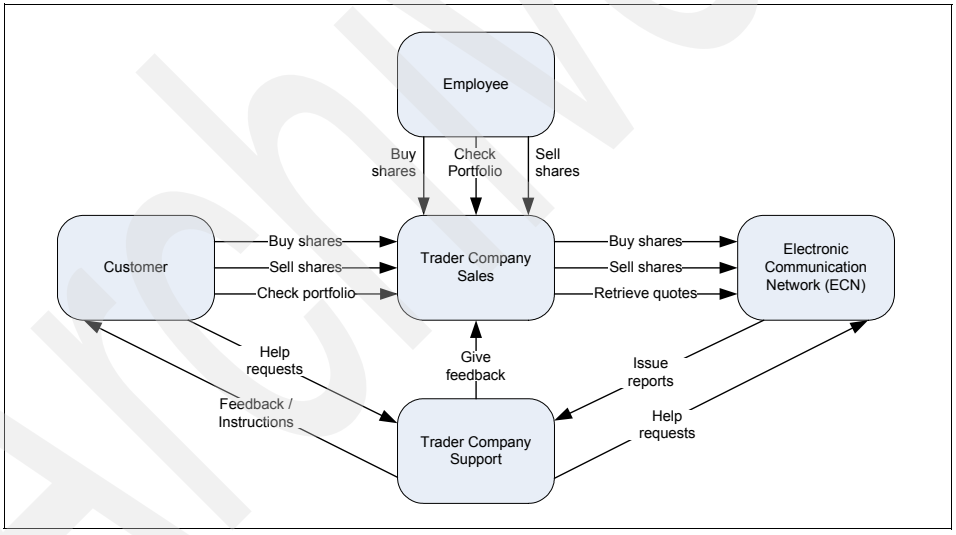


Figure 6-1 Trading company business context diagram

We see the trading company sales department, which primarily interacts with the customers, and processes their buy and sell actions through an electronic communication network (ECN) that is used by financial institutions to exchange shares with the stock exchanges in real time. Additionally, the company allows their employees to trade without commissions and portfolio costs as a benefit. Another department is support, where helps customers with technical issues. This department works close with the sales department to assist customers

having issues with buying or selling shares. They also receive issue reports from the ECN in cases there are maintenance time windows or global constraints that can cause issues.

From an organizational point of view, the trading company is divided into several departments, as illustrated in Figure 6-2. Some of them were already mentioned in the business context diagram, but others have a supporting purpose and are not very relevant for the core business activity of the company.

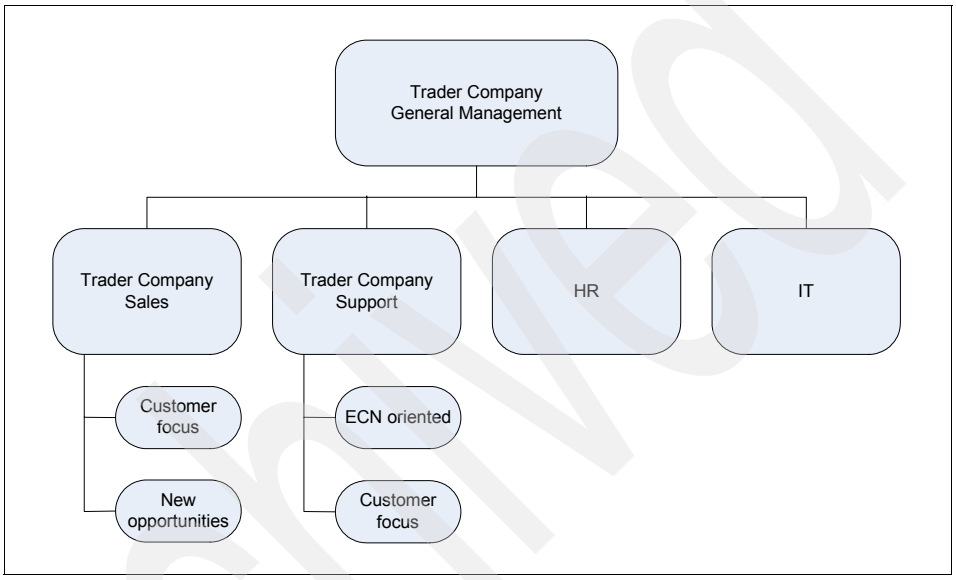


Figure 6-2 Trading company - organizational overview

The company's IT department hosts a simple stock trading application that allows customers and internal users to list the companies, get quotes, and trade stocks of the listed companies. The data is exchanged with the ECN by accessing a database of the current provider directly.

6.1.2 Application view

From a technical perspective, the application is based on a three tier architecture and is hosted within the internal IT department. To provide the maximum amount of user access, the application supports three types of clients residing on the presentation tier: a native Java client, a Web interface, and a Portal based interface. The logic tier is hosted on a J2EE platform and is based on Java EJBs. The trader process implementations are all specialized based on the connection path to the data that has to be accessed on the data tier.

This was necessary to provide access to the continuously changing interfaces of the external ECN and internal sources.

The following connectors are supported by the application to connect to the data tier:

- A CICS ECI J2C resource adapter, providing direct access from Linux to the back-end logic hosted in CICS TS, using the CTG.
- An IMS J2C resource adapter, providing direct access from Linux to the back-end logic hosted in IMS TS.
- A JDBC connection using either straight JDBC from an EJB session or using CMP Entity EJBs, in which case the back-end logic is located with the Web front end and the back-end interface and is implemented as session EJBs.
- A WebSphere MQ JMS provides access to either IMS transactions through the IMS-MQ bridge or CICS terminals through the CICS-MQ bridge.

The connection paths shown in Figure 6-3 shows the current possibilities to integrate with data providers.

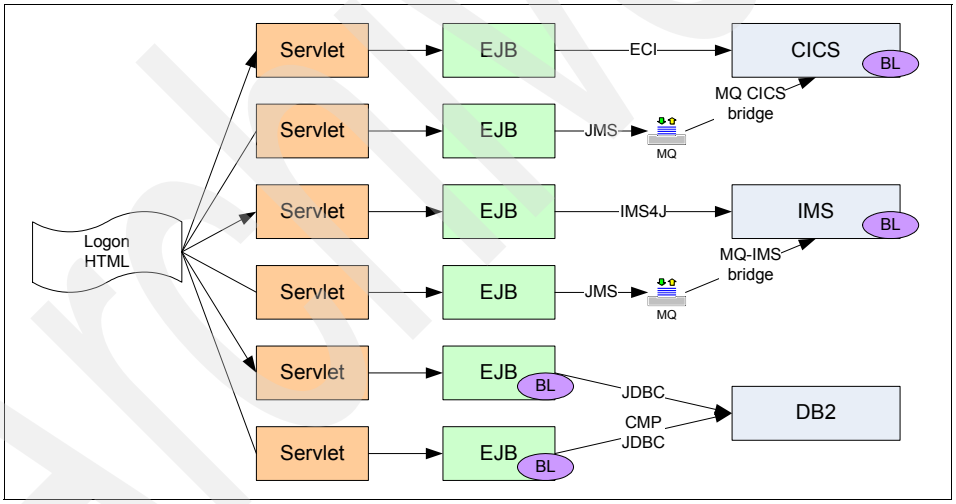


Figure 6-3 Connection paths of the trader application

Even though the connection paths can be seen as generally flexible, the current structure does not contain an abstract layer for accessing the data.

Additionally, the setup consists of UNIX and Linux systems for the server operation with clients based on various other operating systems.

6.2 Using SOA scenarios

Based on the IBM SOA Foundation, we use a selection of the appropriate scenarios we introduce in 2.5.4, “SOA scenarios” on page 61. The use of the SOA scenarios is best represented by using the algorithm shown in Figure 6-4. As the requirements in our example are varied and wide spread across different areas, we show what our specific selection is here.

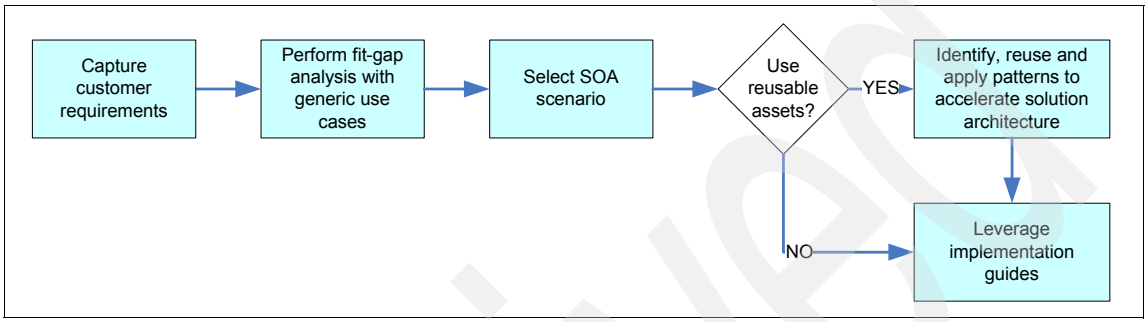


Figure 6-4 Scenario selection process: gather requirements

We use this example to demonstrate the capabilities of System p for the infrastructure layer, the hosted applications, and services by introducing new business needs. Based on these needs, we show how the infrastructure layer can align the IT architecture to support the new needs.

6.2.1 Gather business requirements

Based on the given business and IT views, the demands of the various departments to be more productive guide the selection of our various requirements.

Capturing the requirements is our first step in the process and is illustrated in Figure 6-4. There are several ways to gather requirements, ranging from interviews to more analytical and process driven approaches in the field of requirements engineering.

We will list the requirements gathered from the company and explain them in more detail here:

- IT department: Clear data access and separation

  The current limitation of just having one ECN has lead to a strong dependency on their current standards and systems. In order to connect to the ECN, a continuous effort is necessary to adjust the current communication channels and related data access.

Additionally, access to internal data providers to manage the portfolios must be considered. While trying to develop applications adjusted to this ECN, the internal process for integrating the new modules is an issue. Based on these experiences, the IT department would like more abstract access to the data so that they do not need to rewrite the application logic to keep up with the changes.

- IT department: Easier scaling and management for hosting more accounts

  Because of the continuous growth in the number of customers and employees, the department must continuously add new storage and processing resources to host the solution and associated data. As the number of systems grow, it gets more and more difficult to manage these systems from an administrative perspective. Additionally, the logical view of the data becomes more and more distributed across different systems.

- IT department: Less direct access to the ECN

  The direct access to the ECN requires the IT department to adjust the communication channels and data access systems for this ECN. This effort grows with the number of systems in use and gets more and more difficult for the IT department to manage.

- General management: Interact with external quote providers

  Even though the trading company is one of the fastest offerings on the market right now, the stock exchange quotes provided directly by the ECN are delayed and lack in quality. Management would like to be able to serve customers with a personalized selection of quote providers based on the type of account.

- General management: Be ECN independent

  With more and more competitors in the same field, the trading company would like to benefit from the emerging variety of ECN providers that sometimes offer better conditions and could allow higher earnings.

- General management: Mobile access

  Because cell phones have become a major avenue to access the Web, the customers of the trading company have asked for mobile access to the company's services. Management sees this as an ideal opportunity to differentiate the trading company from its competitors.

- Support department: React proactively to upcoming issues

  Support always acts more reactively than proactively. Issues reported by the ECN are hard to avoid on short notice. Customer complaints about the speed of trades had to reach a specific limit in order to escalate the issue to the IT department to improve the systems.

- Support department: Gain role based access

  As some support queries require interaction with the customer's portfolio, the department has to log in as a customer in order to receive the information needed to give beneficial support. The department would like to have direct access to relevant information about customer portfolios to support them.

- General management: Allow role/consumer specific usage

  Currently, experienced traders, employees, and other individuals share the same system and user interface to the system. Management would like to offer a more role specific offering to incorporate the needs of large scale traders and other businesses. Also, offerings to provide trade services for other companies and allow their departments to manage their employee portfolios are rare and in high demand by the market, which may be another avenue for more revenue.

- General management: Provide further related services

  Through the high dependency of share prices to current news, the management would like to offer additional news feeds to customers.

- HR department: Allow easier handling of employee benefits

  The current process of giving employees benefits in the form of shares is difficult because of the impact of managing additional accounts for these shares. In addition, the nomination and approval process is done manually, which has been inconsistent. The HR department would prefer to add these shares directly to the employees portfolio rather than handing out purpose bound checks. Feedback from managers also indicates that the initiation could be easier.

Addressing all these requirements would require a large scale restructuring of the whole company and its IT infrastructure. In the following sections, we go into details of how we address some of these requirements and show how SOA and System p synergy increase the benefits.

6.2.2 Generic use case selection

We start by selecting the appropriate use cases introduced in 2.5.6, "Use case to scenario mapping" on page 71 based on the customer requirements.

We try to cover most of the business requirements introduced through the use cases. This leads us to the scenario in 6.2.3, "SOA scenario selection" on page 206.

Table 6-1 illustrates how the requirements relate to generic use cases.

Table 6-1 Use case selection

Requirement	Generic use case
IT department: Clear data access and separation	U18: Allow users seamless access to diverse data sources.
IT department: Less direct access to the ECN	
General management: Be ECN independent	
IT department: Easier scaling and management for hosting more accounts	U17: Populate information.
General management: Interact with external quote providers	U8: Enable loose coupling of service consumers and providers using advanced dynamic routing and diverse protocols.
General management: Mobile access	U15: Allow users to access services through client devices.
Support department: React proactively to upcoming issues	U11: Analyze existing business process flows using monitoring.
Support department: Gain role based access	U13: Personalize information based on user profile.
General management: Allow role/consumer specific usage	
General management: Provide further related services	U1: Reuse existing or create new application logic as a service within the enterprise.
HR department: Allow easier handling of employee benefits	U10: Implement a new business process flow.

Based on the SOA entry points introduced in 2.5.5, “SOA entry points” on page 68 and the use cases identified in Table 6-1, we show what the business issues are and what components a solution may consist of from a high level perspective. This can be used in addition to the scenarios to provide a view for executives or general management without getting into too many technical details, as shown in Table 6-2 on page 205.

Table 6-2 Entry point selection

	Business issues	Solution components	Business values
People	► General management: Mobile access (allow broader access based on multiple client devices). ► General management: Allow role/consumer specific usage. ► Support department: Gain role based access.	► Allow diffent account types. ► Abstract the representation from the appplication logic. ► Allow role based data field usage. ► Introduce collaboration features for use within the application. ► Provide a composite portal application to be assembled for the needs of the customer.	► Customer access is independent from the client device. ► Collaboration allows for immediate communication with support and will enhance client satisfaction. ► Assembling the application based on customer needs provides high flexibility for handling a broad range of customer needs and types.
Process	► HR department: Allow easier handling of employee benefits.	► Create an automated process for nominating and approving benefits. ► Integrate the existing portfolios of employees as receivers of these benefits. ► Create interfaces for new types of benefits.	► Reduce the workload associated with the current manual process. ► Increase flexibility for employees by using existing accounts.
Information	► IT department: Clear data access and separation. ► IT department: Less direct access to the ECN. ► IT department: Easier scaling and management for hosting more accounts.	► Store data abstracted from the ECN. ► Create data federation to gain a logic view of data and data accessed from ECNs.	► Leverage the different data across the company. ► Provide independence of information from the ECN.
Connectivity	► IT department: Less direct access to the ECN. ► General management: Be ECN independent. ► General management: Interact with external quote providers.	► Create a common communication channel for data across and beyond the enterprise. ► Open standards based communication to the ECN. ► Open standards based access to external data providers.	► Independency from ECNs. ► Easier integration of external providers by using open standards.

6.2.3 SOA scenario selection

The next step in the process is the selection of matching scenarios based on the generic use cases that have been determined. Based on the tables shown in 6.2.2, “Generic use case selection” on page 203, the following SOA scenarios are related to our requirements:

- Service creation scenario
- Service connectivity
- Interaction and collaboration services
- Business process management scenario
- Information as a service scenario

Additionally, the scenarios of SOA design, SOA governance, and SOA security and management supplement the other scenarios in their implementation.

6.2.4 Identify, reuse, and apply patterns

If reusable assets can be used, patterns will support the implementation further. There is a collection of pattern resources that can be found at:

`http://www-128.ibm.com/developerworks/patterns/`

IBM Redbooks on patterns are also available, for example, *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240.

In addition, you might want to align existing methodologies towards reusable assets, guides, and work products.

6.2.5 Leverage implementation guides

The final step, leveraging implementation guides, includes using working examples that demonstrate how to implement a solution for the phases in the SOA life cycle.

Realizations have been developed to help you understand how a scenario can be used and how products are selected. They are example business cases that describe a real-world industry situation and its solution. There can be multiple realizations for a given scenario. Each scenario includes prescriptive questions that help select you the correct realizations and any decisions that need to be made within the realization.

6.3 The runtime environment

The company tries to implement a service-oriented architecture that can help them to:

- Have more effective integration with Business Partners
- Support customer service initiatives
- Enable employee self service
- Streamline™ the supply chain
- Have more effective use of external service providers
- Facilitate global sourcing

Given the analyzed requirements and a System p for the IT infrastructure, a possible configuration can be shown. As there are several Linux systems existing in the IT environment of the company, some partitions are set up with Linux in order to migrate particular business critical applications to these partitions and allow new features to be developed on them.

We create eight partitions on a single System p server. The partitions are assigned as follows:

- Linux partitions
  - Front end (Bigbend1)
    - WebSphere Application Server Network Deployment (AppClnt02)
  - Mediation (Bigbend2 and Bigbend3)
    - WebSphere Enterprise Service Bus (ESB02)
    - WebSphere Message Broker (Linux_Broker)
  - Back end (Texas)
    - WebSphere Application Server Network Deployment (AppSrvr2)
- AIX partitions
  - Front end (Alamo2)
    - WebSphere Application Server Network Deployment (AppClnt01)
  - Mediation (Houston1 and Alamo1)
    - WebSphere Enterprise Service Bus (ESB01)
    - WebSphere Message Broker (AIX_Broker)
  - Back end (Houston2)
    - WebSphere Application Server Network Deployment (AppSrvr1)

Figure 6-5 shows the diagram of the environment that we implement.

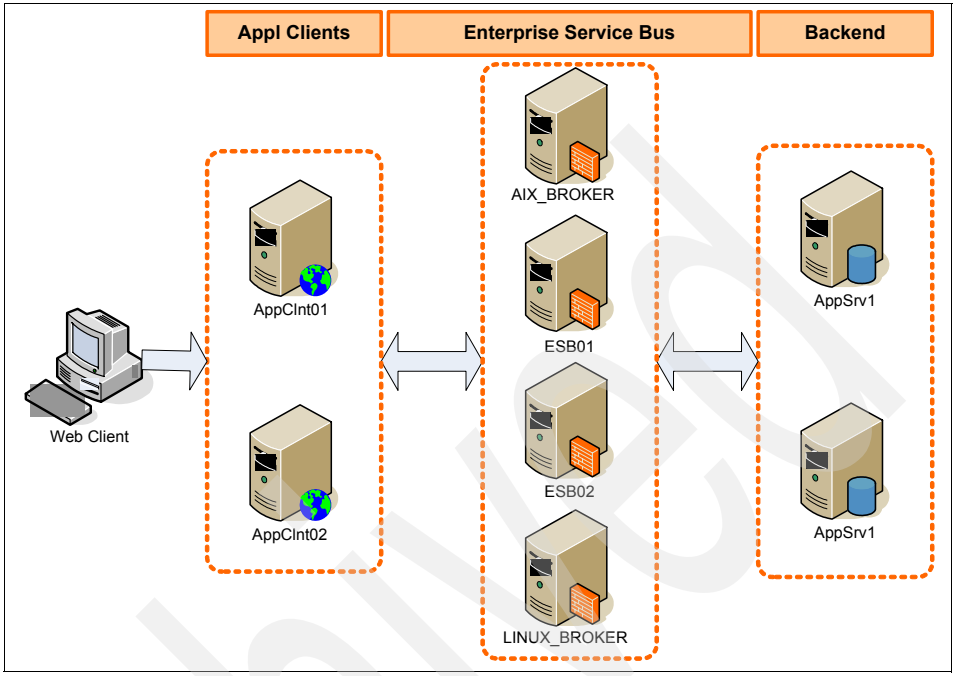


Figure 6-5 SOA scenario on System p

6.3.1 Front-end J2EE Web application

The front-end Web application is developed using the WebServices client wizard Trader Service Projects. The application consists of:

- Initial login page in login.html
- ListCompanyServlet (Invokes the back-end ListCompany WebServices.)
- GetQuotesServlet (Invokes the back-end getQuote WebServices)
- BuySellServlet (Invokes either the buy or WebService.)
- logoutSellServlet (Clears up the session bean.)

We provide three types of enterprise application archive (ear) files for the client interface:

- TraderClientEAR: This ear file runs the TraderClientWeb application that provides the basic trader application functionality.
- TraderClientMemEAR: This ear file runs the TraderClientMem application that has a memory leak in the logic for testing a memory leak situation.

- TraderClientLckEAR: This ear file runs the TraderClientLck application that has a lock problem injected for testing a dead lock situation.

6.3.2 Back-end implementation

The back-end system consist of two entities: the company and the customer. The company has the quotes definitions, and the customer databases have the customer's name and stock ownership.

6.3.3 Mediation implementation

We discuss the implementation for each ESB that we use.

WebSphere Enterprise Service Bus mediation

The WebSphere Enterprise Service Bus mediation that we use is simple mediation logic that queries the WebSphere Services Registry and Repository (WSRR) server for the location of the service and invokes it. This allows the injection of logic to check which Web services servers are available and redirect the call if necessary.

WebSphere Message Broker mediation

The message broker implementation is currently using a hardcoded target. Further enhancement may be done to connect WebSphere Message Broker to WebSphere Services Registry and Repository to query the Web services endpoint. We do not do that in this project. The broker logic uses four message flow definitions stored in a broker archive (bar) file. The bar file is then deployed to the message broker execution group for processing the Web services calls.

The message flow logic uses the following items:

- An HTTP input node that gets the Web services requests.
- The input node that forwards the request to the HTTP request node that performs the actual Web services call. The endpoint address is hardcoded here.
- The HTTP reply node that sends the result back to the requester.

6.3.4 Resources structure of the setup

Based on the setup shown in Figure 6-5 on page 208, the resources can be provisioned based on the structure indicated by the rectangles surrounding them. The applications could be partitoned on their own, meaning one partition for each application server, for example, or could be installed on a single partition. You need to decide on the granularity of fine tuning required.

- Application server partition(s)

  The application servers are kept in a single partition to enable allocation of resources based on the load that is being expected from a client side perspective.

  The use of uncapped micro partitions in conjunction with a shared processor pool is best suited for high peaks in requests being handled efficiently by the system. Another possibility is to use WPARs for each application server and assign the resource dedication of the partition based on the importance or performance requirements of the access method hosted by the server.

  In regards to the main memory, the partition will be dependent on the amount used by the application consumption of the heap.

  Regarding hard disk space, the partitions will not require a very flexible sizing; just keep the amount of logging in mind when the size is determined.

- ESB partition(s)

  A major requirement for the middleware is scaling and interconnecting with the external service providers. Additionally, the load peaks that can be caused by the web clients will be passed through in order to access data from the back-end services. This requires some dedicated resources in conjunction with a shared processor pool in order to achieve ensured availability for external providers and the ability to handle high peaks of requests.

- Back-end partiton(s)

  As the back end contains the databases accessed from the middleware, the major requirement is scaling based on hard disk size. Additionally, a continuous load might exist based on communications with external data providers or systems.

6.4 Application feedback and monitoring

This section discusses some additional monitoring capabilities that are available on System p environment. The performance of the running system must be monitored continuously. A dynamic environment, such as SOA, requires a structured approach to monitoring performance and anticipating changes. We discuss monitoring with the following tools:

- 6.4.1, “IBM Director” on page 211
- 6.4.2, “IBM Tivoli Monitoring” on page 214
- 6.4.3, “Application management tool” on page 217

6.4.1 IBM Director

IBM Director is a platform management family that provides the tools for coordinating and managing all of your virtual and physical resources in the data center. The tools are used in conjunction with the hardware infrastructure of IBM servers. Figure 6-6 shows the system list from the IBM Director console. It shows the systems that it manages.

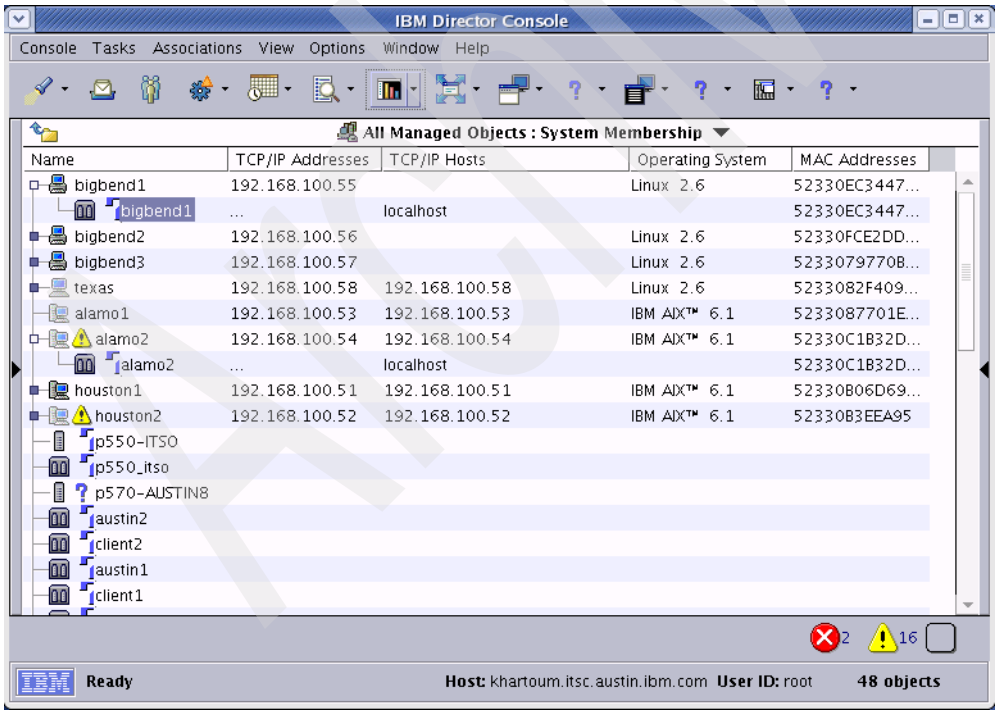


Figure 6-6 IBM Director

The managed systems in IBM Director can use the IBM Director agent (with detailed Operating System information) or agentless (SNMP based).

Figure 6-7 shows tasks that can be operated on a partition. It shows the tasks that we use with the virtualization manager plug-in. You can start, stop, and perform virtualization actions from IBM Director.

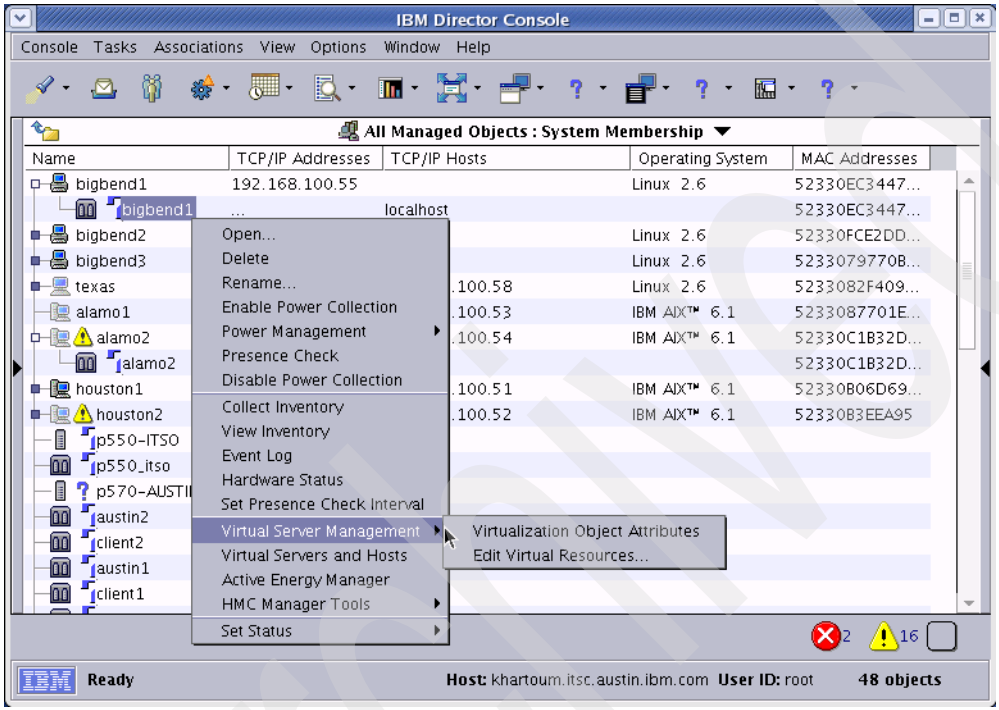


Figure 6-7 IBM Director facility

Other facilities can be accessed from the Tasks menu. Figure 6-8 on page 213 shows these facilities. These are the main actions that can be performed:

- Resource monitoring: Collecting system information, such as processor, memory, and IO rates.
- Server configuration and inventory: Collecting server configuration and storing the information in a database for reporting.
- Software update and deployment: Installing software and patches to managed systems.
- Access to system: Using terminal mode for Linux or AIX.

- Additional plug-in facilities that we install are:
  - Active Energy Manager
  - Virtualization Manager
  - HMC Manager

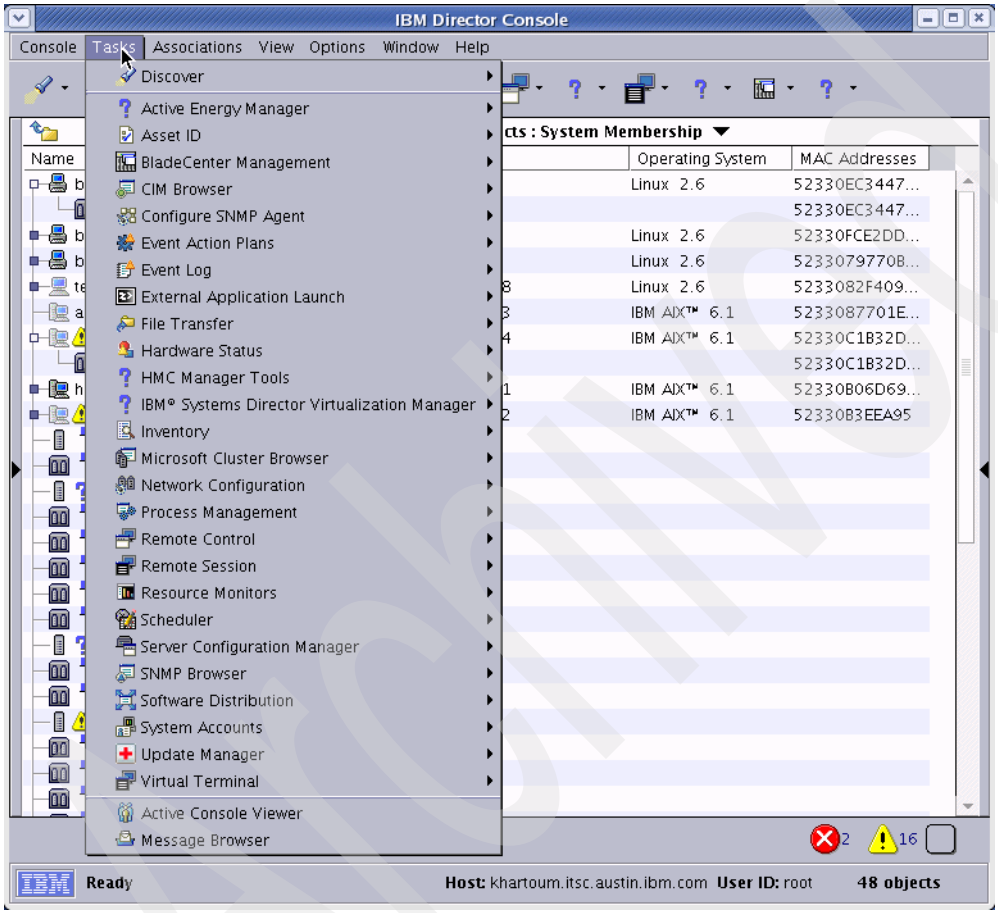


Figure 6-8 Additional tasks

IBM Director provides a comprehensive systems management tools for the operating system and server level. It ranges from inventory, software installation, and resource monitoring.

6.4.2 IBM Tivoli Monitoring

IBM Tivoli Monitoring is a comprehensive performance monitoring tool for systems, middleware, and applications. It provides agents for the operating systems level, middleware platforms, and application monitoring. The console is called Tivoli Enterprise Portal. Figure 6-9 shows a sample portal display.

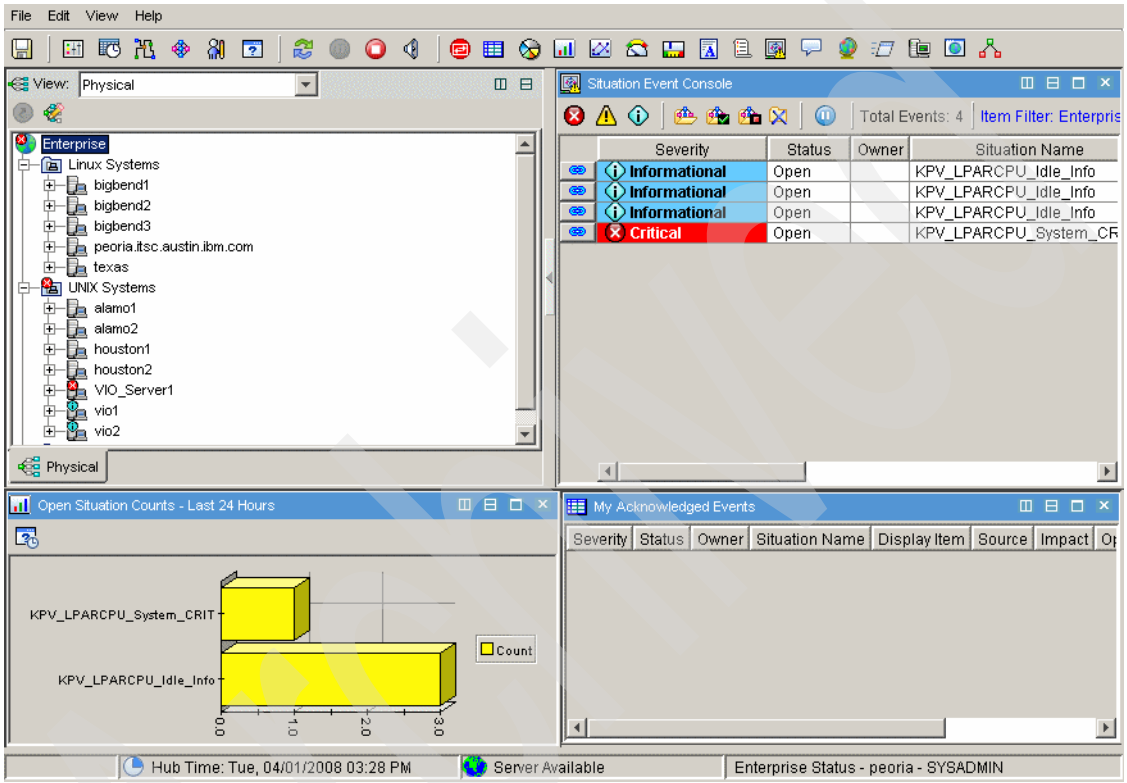


Figure 6-9 Tivoli Enterprise Portal

The managed systems are arranged based on their operating systems. The enterprise display shows general events for the whole enterprise. The agents are shown under each servers.

Figure 6-10 shows a sample Linux operating system monitoring page. It contains information related to the system resources usage of the Linux operating system, including processor, memory, and IO information. Under the Linux OS branch, other workspaces exist that deal with individual metrics, such as CPU, memory, and processes performance.

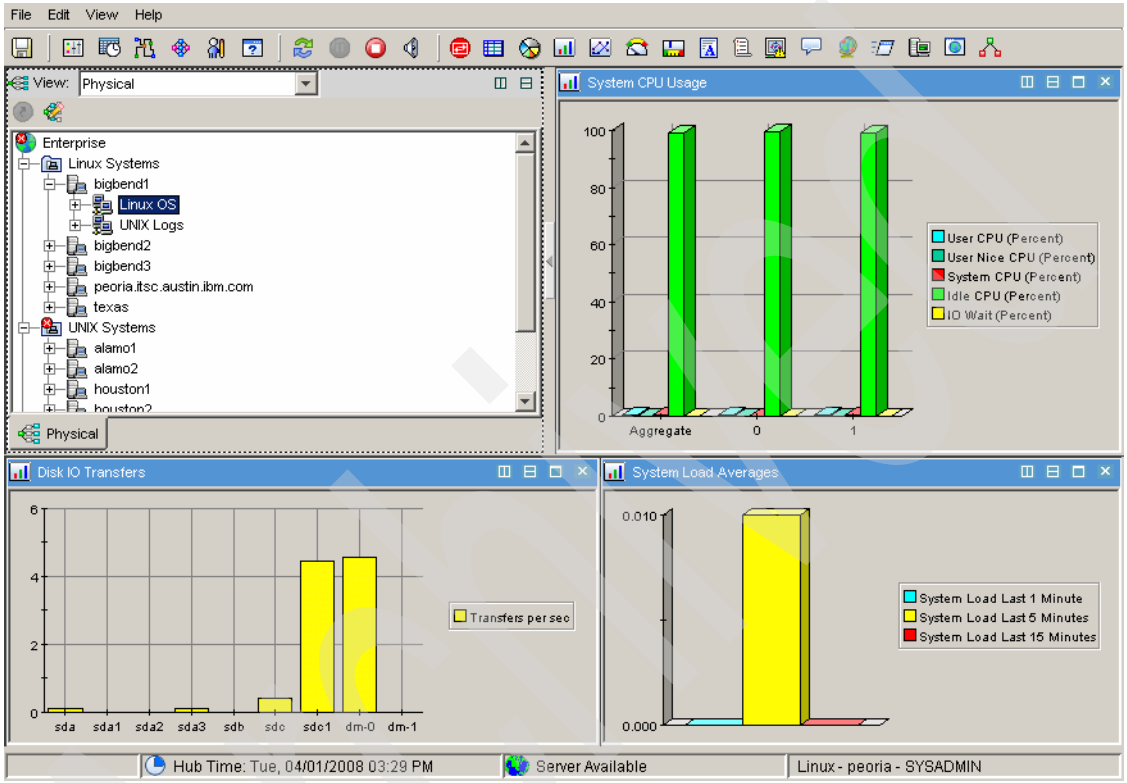


Figure 6-10 Linux monitoring

Figure 6-11 shows the workspace for AIX premium monitoring. This monitoring tool understands virtualization metrics and other System p related resources. It can collect VIOS and HMC information.

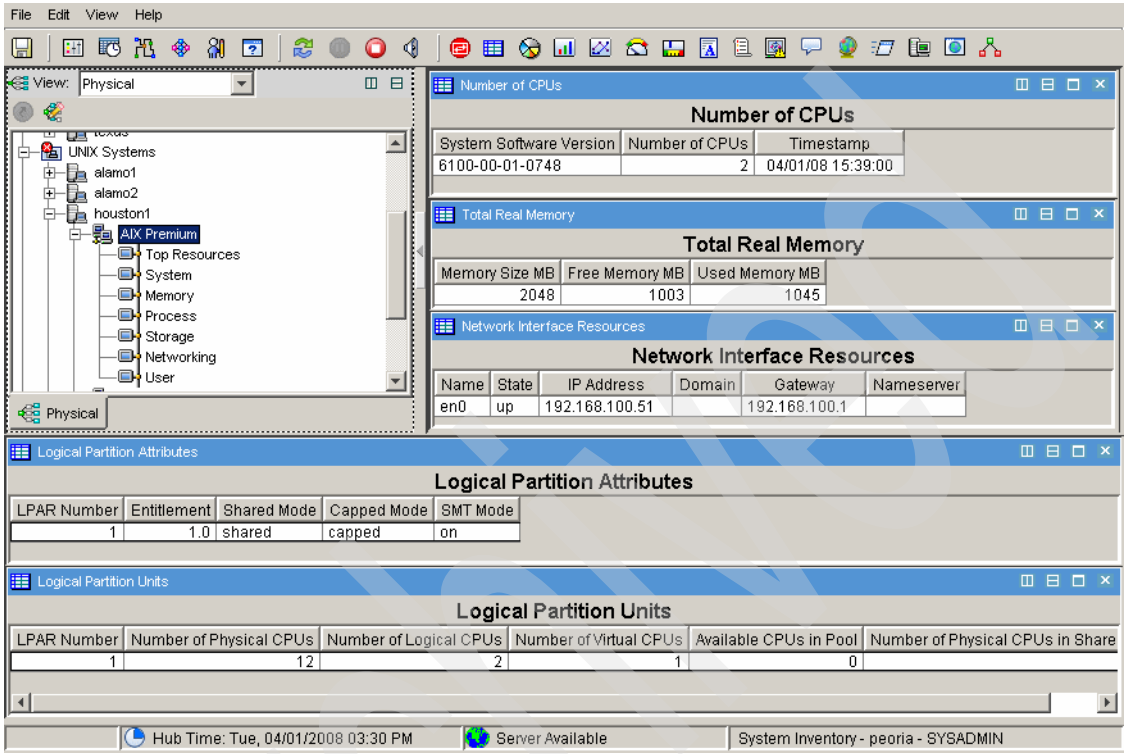


Figure 6-11 AIX premium monitoring

Figure 6-12 on page 217 shows a sample Top Resources workspace for an AIX partition.

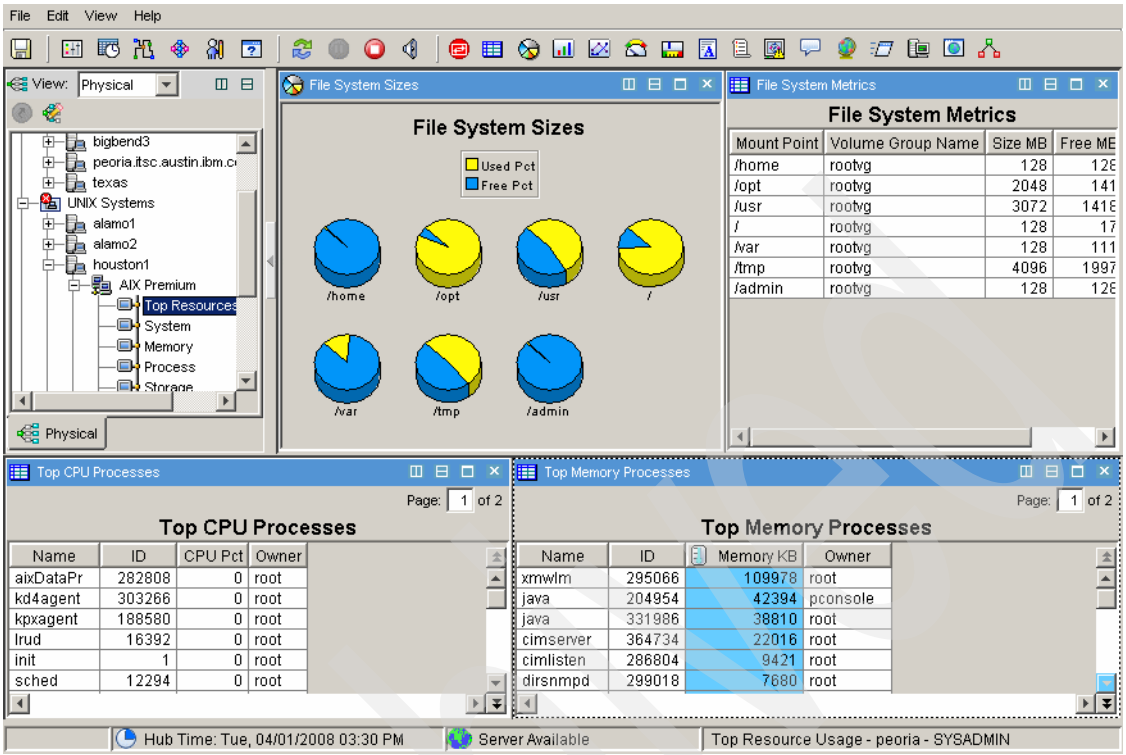


Figure 6-12 AIX monitoring

IBM Tivoli Monitoring is primarily used to collect performance and capacity information regarding a running system. It provides system specific metrics regarding processor, memory, and IO usage. IBM Tivoli Monitoring can be extended to include the application management tool from the IBM Tivoli Composite Application Manager family.

6.4.3 Application management tool

In a SOA environment, it is necessary to monitor, manage, and analyze the deployed application and transactions to enable IT service management.

IBM Tivoli Composite Application Manager (ITCAM) is a family of products that is part of the application management building block of the IBM Tivoli portfolio. The system management portfolio of IBM Tivoli consists of a set of systems management suites to manage your entire IT infrastructure.

ITCAM has three products that are relevant to the SOA management: ITCAM for SOA, ITCAM for WebSphere, and ITCAM for Response Time Tracking.

ITCAM for SOA

IBM Tivoli Composite Application Manager for SOA (ITCAM for SOA) is a product based on IBM Tivoli Monitoring V6.1. It monitors, manages, and controls Web services and service-oriented architectures (SOAs) deployed using a wide range of IBM and third-party systems.

ITCAM for SOA recognizes and quickly isolates Web service performance problems, alerts you when Web service performance is degraded, and reports results against committed service levels. It provides an integrated, easy-to-use console that helps you visualize the flows of Web services in their entirety. It monitors your services where you want them, with heterogeneous platform support and views by service requestor support, and reports on the number of requests or response times by the requestor.

ITCAM for SOA includes the Web Services Navigator, a plug-in to IBM Rational and other Eclipse-based tools, which provides a deep understanding of service flows, patterns, and relationships to developers and architects using operational data from the Tivoli Data Warehouse or monitoring log files. ITCAM for SOA is a core component of the IBM SOA Foundation Management Essentials, an integrated and open set of software, best practices, patterns, and skill resources to get you started with service-oriented architectures.

ITCAM for SOA Version 6.1 with Fix Pack 1 supports:

- WebSphere Application Server
- WebSphere Enterprise Service Bus
- WebSphere Process Server
- WebSphere Message Broker
- WebSphere DataPower Appliance

It also gives support to several application and transaction servers.

For more information about ITCAM for SOA, go to:

http://www.ibm.com/software/tivoli/products/composite-application-mgr-soa/

ITCAM for WebSphere

IBM Tivoli Composite Application Manager for WebSphere (ITCAM for WebSphere) is an evolution from WebSphere Studio Application Monitor and OMEGAMON® XE for WebSphere Application Server.

ITCAM for WebSphere enables you to analyze the health of the WebSphere Application Server and the transactions that are invoked in it. It is able to trace the transaction execution to the detailed method-level information, and connects transactions that spawn from one application server and invokes services from other application servers, including mainframe applications in IMS or CICS.

ITCAM for WebSphere provides a flexible level of monitoring, from an non-intrusive production ready monitor, to a detailed deep-dive tracing for problems of locking or even memory leaks. ITCAM for WebSphere provides a separate interactive Web console and also allows monitoring data to be displayed on the Tivoli Enterprise Portal.

ITCAM for WebSphere Version 6.1 provides additional functions, such as:

- Integration with IBM Tivoli Service Manager by providing a Web services interface to get health status
- Improved memory leak and locking analysis pages
- Problem determination enhancements
- Advanced visualization, aggregation, persistence, and correlation of performance metrics in Tivoli Enterprise Portal
- Additional WebSphere server platform support, including WebSphere Portal Server and WebSphere Process Server
- Enhanced composite transaction tracing and decomposition
- Web session browser to help diagnose session-related problems

For more information, see the ITCAM for WebSphere page at:

http://www.ibm.com/software/tivoli/products/composite-application-mgrwebsphere/

ITCAM for Response Time Tracking

IBM Tivoli Composite Application Manager for Response Time Tracking (ITCAM for Response Time Tracking) is an evolution from IBM Tivoli Monitoring for Transaction Performance, Candle® End-to-end Watch, and Web Response Monitor.

ITCAM for Response Time Tracking allows monitoring and analysis of application transaction response time. It provides statistics of response times using instrumentation and robotic means. ITCAM for Response Time Tracking enables you to analyze and break down response time into individual components to quickly pinpoint a response time problem.

ITCAM for Response Time Tracking can decompose transactions by robotic means that simulate users, tracking its execution in J2EE application servers all the way to the IMS or CICS back end. The response time information is presented on the Web management console or Tivoli Enterprise Portal.

For more information about ITCAM for Response Time Tracking, go to:

http://www.ibm.com/software/tivoli/products/composite-application-mgr-rtt/

A

WebSphere Application Server and WPAR

WebSphere Application Server Version 6.1 is supported inside a system workload partition. It can be installed either from the global environment or from within the workload partition. We discuss the following topics:

- "Considerations" on page 222
- "Creating workload partitions" on page 223
- "WebSphere Application Server installation wizard" on page 223
- "Silent installation with profile creation" on page 224
- "Profile management" on page 225
- "Update installer" on page 225
- "WebSphere feature packs installation" on page 226

Considerations

The information in this appendix comes from the Technotes found at:

http://www-1.ibm.com/support/docview.wss?rs=180&context=SSEQTP&dc=DB520&dc=DB560&uid=swg21291889&loc=en_US&cs=UTF-8&lang=en&rss=ct180websphere

and

http://www-1.ibm.com/support/docview.wss?rs=180&context=SSEQTP&dc=DB520&dc=DB560&uid=swg21293695&loc=en_US&cs=UTF-8&lang=en&rss=ct180websphere

There are some installation items to consider:

- Installation location

  WebSphere Application Server components are typically installed under the /opt or /usr directory. For a system workload partition, the default is to have a shared read-only /opt and /usr directories. The -l option copies the /opt and /usr directories under the workload partitions.

  If the /opt and /usr paths are shared, then WebSphere can be installed under the /home directory of the workload partition (or from the /wpars/wpare_name/home directory of the global partition). Otherwise, it can be installed under /usr.

  Note: If you already have WebSphere Application Server installed on the global partition under /usr, creating a workload partition will place WebSphere Application Server in the partition. Do not use that WebSphere Application Server installation; because it was not installed through the standard installation process, the host name settings and other information are not correct. Use the **`uninstall`** command to uninstall that copy of WebSphere Application Server.

- File system size

  The target installation path of the WebSphere Application Server will not be adequate for installing the server. Depending on where you install WebSphere Application Server, you may need to increase the /wpars/wpar_name/home or /wpars/wpar_name/usr using the **`smitty manfs`** command.

  Select **Enhanced Journaled File Systems** → **Change / Show Characteristics of an Enhanced Journaled File System**. Select the file system that you want to change and increase the number of units for the SIZE parameter so that you have at least several gigabytes of disk space allocated for that file system.

Creating workload partitions

To create a system workload partition, you must be logged in as the root user and run the **mkwpar** command. The workload partition name must be a network addressable entity, either using /etc/hosts or DNS lookup. You create the workload partition using the following command:

```
mkwpar -r {-l} -n wpar_name
```

The -l option indicates a non-shared /usr and /opt directory. All contents the of /usr and /opt directory from the global partition would be copied to the workload partition file system. The workload partition has its own file system under /wpars/wpar_name.

You can start the workload partition using the command:

```
startwpar wpar_name
```

WebSphere Application Server installation wizard

Installing WebSphere Application Server from the wizard can be performed from the global environment or from within the workload partition. Running the wizard from the global environment has the disadvantage that some environment values are incorrect, such as:

- Host name of the target partition is different than the global environment.
- The target directory path must be prefixed with /wpars/wpar_name, such as /wpars/wpar_name/home/IBM or /wpars/wpar_name/usr/IBM.
- Profile creation is not possible, as you cannot respecify the host name of the target partition.

Running the wizard from within the workload partition allows you to have the standard host name and path. However, the prerequisite check and disk space checking has to be disabled. Additional options must be used.

- Use these options for application server installation:
  - -W checklateprereqs.active=False
  - -W lateprereqsfailedpanelInstallWizardBean.active=False
  - -W checklateprereqs.prereqsPassed=True
  - -W calculatediskspaceInstallWizardBean.active=False

- Use these options for other components:
  - -W checkprereqs.active=False
  - -W prereqsfailedpanelInstallWizardBean.active=False
  - -W checkprereqs.prereqsPassed=True
  - -W calculatediskspaceInstallWizardBean.active=False

The installation target directories for WebSphere Application Server components are:

- <path_prefix>/WebSphere/AppServer
- <path_prefix>/HTTPServer
- <path_prefix>/HTTPServer/Plugins
- <path_prefix>/AppClient
- <path_prefix>/UpdateInstaller

The installation wizard invocation can be invoked directly using the **install** shell script command for 64-bit installation. For a 32-bit installation, there is a problem with the script. You can download an updated script from the URLs shown in “Considerations” on page 222 or invoke the **java** command directly. The **java** command has to be invoked from within the component directory. The command is similar to:

```
../JDK/jre.pak/repository/package.java.jre/java/jre/bin/java -cp
setup.jar run <arguments>
```

Silent installation with profile creation

Here are the silent installation options in the responsefile.txt file that must be modified for creating profiles within the WPAR:

- silentInstallLicenseAcceptance=“true”
- disableOSPrereqChecking=“true”
- installType= “installNew”
- installLocation= “<installation_location>/WebSphere/AppServer”
- PROF_hostName=wpar_host_name

Note: The installation location depends on whether you invoke the command from the global environment (/wpars/wpar_name) or from within the partition. It is depends on whether you have a shared installation (use home) or a non-shared installation (use usr). So the possible combinations are:

- /wpars/wpar_name/home/IBM (from global environment, shared)
- /wpars/wpar_name/usr/IBM (from global environment, non-shared)
- /home/IBM (from within, shared)
- /usr/IBM (from within, non-shared)

Silent installation of the WebSphere Application Server component can be invoked using the following command:

```
./install -options /tmp/WAS/your_options_file.txt -silent
```

For additional information about performing a silent installation, read the Installing silently topic in the Information Center for WebSphere Application Server Version 6.1.

Profile management

Profile management is performed using the **manageprofiles.sh** command. Defining and managing profiles must be performed with care for the following reasons:

- Host name differences from the global environment and partitions. The -hostName option should reflect the correct host name where the profile would be used.
- Profile path differences from the global environment and partitions. The -profilePath option should reflect the correct path relative to the environment you are running it from. You must use the /wpars/wpar_name prefix if running from the global environment.

Update installer

When you use a non-root user to install the Update Installer, you might encounter a Non-root User Limitation panel with the following message:

```
The AIX user account running the Update Installer program also must be
able to run the slibclean command; otherwise, a root user must run the
slibclean command before the Update Installer program runs.
```

If you attempt to run the **slibclean** command from within the WPAR as a non-root user, the command might return the error message:

```
bash: /usr/sbin/slibclean: The file access permissions do not allow the
specified action.
```

Before you install the Update Installer from within the WPAR, you must grant the appropriate permission to /usr/sbin/slibclean to allow that command to be run by non-root users. Use the following command to change the permissions:

```
chmod 555 /usr/sbin/slibclean
```

If the WPAR is non-shared, it has private /usr and /opt file systems. It is not a problem to change the mode from within the WPAR. However, if the WPAR has shared /usr and /opt file systems, you cannot change the mode of /usr/sbin/slibclean from within the WPAR. In that case, you must run the **chmod** command from the global environment.

WebSphere feature packs installation

You can install the WebSphere Application Server Version 6.1 feature packs to a WPAR installation of WebSphere Application Server. To successfully install the Feature Pack for Web Services to a WPAR installation of WebSphere Application Server, run the **install** command from the global environment rather than from within the WPAR.

B

Workload partitions benefits

Workload partitions offer new possibilities for managing AIX environments. They complement other virtualization solutions available for System p platforms. The following sections show the various benefits of using workload partitions:

- “Improvement of service level agreements” on page 228
- “Optimization of resource usage” on page 228
- “Highly granular control of resource allocation” on page 229
- “Control of security and privilege commands” on page 229
- “Simplified handling of software stack” on page 230

Improvement of service level agreements

Hardware components of an IT infrastructure might need to undergo maintenance operations requiring the component to be powered off. If an application is not part of a cluster of servers providing continuous availability, either for technical, organizational, or cost reasons, WPARs can help reduce the application downtime. Using the Live Partition Mobility feature, the applications that execute on a physical server can be temporarily moved to another server without an application blackout period during the period of time required to perform the server physical maintenance operations. Long running jobs can take advantage of the checkpoint/restart feature of WPARs. This feature can be used to protect the jobs against a failure that requires restarting all computations from the beginning. The checkpoint feature can be used to regularly capture a snapshot of the application runtime environment, without having to instrument the code. In the case where the job needs to be stopped before reaching completion, the job can be resumed in the state it was in when the last checkpoint was saved. The checkpoint/restart feature can also be used to execute long lasting batch jobs on a system with limited resources. This job can be run at night time, be paused during the day time, when the computer resources have to be dedicated to other applications, such as transaction handling or Web serving, and then resumed at the beginning of the next night.

The workload partition technology can also help in an environment where an application needs to be started often, on demand, and quickly. This might apply, for example, in test environments where resources are too scarce to keep multiple applications executing concurrently when not in use. Using WPARs, many applications can be defined on a server, but not activated. Activation of the workload partitions executing each of these applications can be performed only when needed for a test.

Optimization of resource usage

WPAR technology gives you additional flexibility in system capacity planning as part of a strategy for maximizing system utilization and provisioning efficiency.

Due to the static allocation of partitions in physical servers, in a typical IT environment, each server is sized with spare capacity to allow for a resource consumption increase of all applications executing within this server. Due to the mobility feature of WPARs, the server sizing and planning can be based on the overall resources of a group of servers, rather than being performed server per server. When an application grows and requires resources that can no longer be provided by the server, the application can be moved to a different server with spare capacity.

The same mobility feature, combined with the policy-based relocation functions of the WPAR Manager, allows you to size a set of servers to handle the peak load, based on the overall resource capacity of the set of servers, and not for each server. In a classic environment, each server must be able to support the peak load of all partitions hosted within that server. With application mobility, it is possible to take advantage of free resources in one physical server to offload another physical server hosting applications that require more resources than are locally available. AIX V6 provides highly granulated control of CPU and memory resource allocation to workload partitions (down to 0.01% increments). This technology is therefore suitable for server consolidation of very small workloads. This can be particularly interesting for the replacement of old servers, for which even 10% of one POWER5 or POWER6 processor (the smallest micro partition) exceeds the application needs.

Highly granular control of resource allocation

When multiple applications are executing within the same AIX instance, the system administrator might want to control the amount of CPU and memory resources used by each application. One way to perform this control is to set up the Workload Manager (WLM) functions that are part of standard AIX features. The WPAR technology provides a new way to perform this resource control. The WPAR resource control reuses the WLM technology, but encapsulates it in a way that WLM is invisible to the system administrator. There is no need for the system administrator to know about WLM. The resource control is available through options of the WPAR command-line and SMIT interfaces. The WPAR resource control feature allows the system administrator to arbitrate between applications competing for CPU and memory resources. This guarantees that each application receives a share of the CPU and memory resource available from the global environment. These resources are separate from the requirements of the other applications executing in WPARs within the same operating system instance.

Control of security and privilege commands

In large AIX environments, where a partition hosts many applications, it is not unusual to have multiple people acting as system administrators. However, all of them might not need root or superuser privileges in all domains of system administration. These people can be specialized for activities, such as user administration, network control, storage control, or software maintenance.

WPAR technology supports this specialization of roles and can help restrict the privileges given to one person to just the scope that person needs to control. System workload partitions have their own user set, independent from the user set defined at the global environment level. An individual, who is using root within a system workload partition, only has superuser privileges for the resources visible within this WPAR. This user cannot control global environment resources, such as network adapter or physical devices, and cannot act on resources belonging to other workload partitions. Many applications need the application administrator to use the root user to control the application, even if this person does not need to manage the operating system. WPAR technology allows you to delegate the superuser privileges to one individual and limit them to an application environment without jeopardizing the global environment.

The separation of user sets (or security domains) between different system workload partitions also enables the system administrators to isolate groups of users logging on in AIX environments according to their application access control requirements. Users defined in one system WPAR are unaware of the applications executing in the global environment or in other WPARs. They cannot see the list of users or processes outside their WPAR.

Simplified handling of software stack

The WPAR technology can help the system administrator simplify the way that the system administrator maintains the operating systems and application software stacks. For a long time, the traditional approach to application deployment has been to dedicate one server to one application. With the advent of virtualization and partitioning technologies, it has been possible to host multiple applications within partitions of a physical server. But this solution still implies that the system administrator needs to maintain one operating system instance for each application. The WPAR technology allows you to share an AIX instance between multiple applications, while still running each application within its own environment, providing isolation between applications. In this case, the more applications that are consolidated within one AIX instance, the less the system administrator has to perform OS fix applications, backups, migration, and other OS maintenance tasks. However, note that this type of consolidation requires that all applications can run under the same version and maintenance level of the OS.

In addition to sharing the operating system, the system administrator can take advantage of the WPAR technology to share application code. In a traditional AIX environment, if several Apache Web servers are needed, they each need to be deployed in a dedicated server or LPAR. In a WPAR environment, it is possible to install Apache in one LPAR and then execute multiple instances of the Apache

server within this LPAR, by starting multiple WPARs. Each WPAR runs its own Apache server with its own data in a dedicated disk space, but shares the Apache code with all other WPARs. This type of a configuration optimizes memory utilization by eliminating duplication of code and reduces administration maintenance of the Apache code, which only needs to be updated once for all server instances.

IBM AIX Version 6.1 introduces a new concept in software installation and management: *relocatable software packages*. A relocatable application is an application where the files can be installed relative to a base directory that is different from the root directory of the AIX environment. Using this feature, it is possible to deploy multiple versions of the same application within one AIX instance. The system administrator can take advantage of relocatable applications by starting each version of the application in a specific WPAR, therefore providing multiple servers with different server code versions from one LPAR.

The workload partition configuration can be stored in human-readable specification files. These specification files can be generated by the operating system from preexisting workload partitions or can be edited, created, or modified manually. In an environment where a system administrator has to manage several application environments, the WPAR technology can help the system administrator quickly clone and define new application environments. These specification files can be used as input to WPAR creation commands, allowing the system administrator to automate the startup and handling of multiple workload partitions through scripts and programs.

Abbreviations and acronyms

ACL	Access Control List
AES	Advanced Encryption Standard
AIX	Advanced Interactive eXecutive
API	Application Programming Interface
ARM	Application Response Measurement
BPEL	Business Process Execution Language
BPM	Business Process Modeling
CAPP	Controlled Access Protection Profile
CD	Compact Disk
CICS	Customer Information Control System
CMP	Container Managed Persistent
COBOL	Common Business Oriented Language
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
CSM	Cluster System Management
CSV	Comma Separated Value
CTG	CICS Transaction Gateway
DLPAR	Dynamic Logical Partitioning
DMZ	Demilitarized Zone
DNS	Domain Name Service
DRAM	Dynamic Random Access Memory
DVD	Digital Video Disk
ECC	Error Correction Code
ECI	External CICS Interface
ECN	Enterprise Communication Network
EDI	Electronic Data Interchange
EFS	Encrypting File System
EIS	Enterprise Information System
EJB	Enterprise Java Beans
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
EWLM	Enterprise Workload Manager
FFDC	First Failure Data Capture
FTP	File Transfer Protocol
GB	Gigabyte
GUI	Graphical User Interface
HA	High Availability
HACMP	High Availability Cluster Management Program
HBA	High Byte Count
HMC	Hardware Management Console
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transport Protocol
I/O	Input Output
IBM	International Business Machine Corp.
IEEE	Institute Electrical Electronics Engineers
IIOP	Internet Interoperability Object Protocol
IMS	Information Management System
IP	Internet Protocol
IPC	Inter Process Communication

ISO	International Standard Organization
ISV	Independent Software Vendor
IT	Information Technology
ITCAM	IBM Tivoli Composite Application Manager
ITM	IBM Tivoli Monitoring
ITSM	IBM Tivoli Service Management
ITSO	International Technical Support Organization
IVM	Integrated Virtualization Manager
IVR	Interactive Voice Response
JCA	Java Connectivity Architecture
JDBC	Java Database Connectivity
JMS	Java Messaging Services
LAN	Local Area Network
LDAP	Light-weight Directory Access Protocol
LPAR	Logical Partition
MAC	Media Access Control
MQ	Messaging and Queuing
NFR	Not for Resale
NFS	Network File System
NIM	Network Installation Manager
NSA	National Security Agency
OGSA	Open Grid Services Architecture
OS	Operating System
PAM	Pluggable Authentication Module
PC	Personal Computer
PCI	Peripheral Component Interconnect
RAS	Reliability, Availability, and Scalability
RDS	Relational Data Source
RMC	Resource Monitoring and Control
RMI	Remote Method Invocation
ROI	Return on Investment
RPM	Red Hat Package Manager
RSCT	Reliable Scalable Clustering Technology
SAML	Security Assertion Markup Language
SAN	Storage Area Network
SCA	Service Component Architecture
SCSI	Small Computer System Interface
SEA	Shared Ethernet Adapter
SGMM	SOA Governance and Management Method
SLA	Service Level Architecture
SME	Subject Matter Expert
SMIT	System Management Interface Tool
SMP	Symmetrical Multi Processor
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SOMA	Service-Oriented Modeling Approach
SQL	Structured Query Language
TCP	Transmission Control Protocol
UDDI	Universal Description Discovery and Integration
UML	Universal Markup Language
URL	Universal Resource Locator
VIO	Virtual IO

VIOS	Virtual IO Server
VLAN	Virtual LAN
VM	Virtual Machine
WLM	Workload Manager
WPAR	Workload Partition
WS	Web Services
WSDL	Web Services Description Language
WSDM	Web Services Distributed Management
WSRP	Web Services For Remote Portlets
WSRR	WebSphere Service Registry and Repository
XML	eXtensible Markup Language
XSD	XML Schema Definition
XSLT	eXtensible Stylesheet Language Transformations

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 240. Note that some of the documents referenced here may be available in softcopy only.

- *End-to-end Automation with IBM Tivoli System Automation for Multiplatforms*, SG24-7117
- *Getting Started with PowerVM Lx86*, REDP-4298
- *Hardware Management Console V7 Handbook*, SG24-7491
- *IBM Director on System p5*, REDP-4219
- *IBM System p Advanced POWER Virtualization (PowerVM) Best Practices*, REDP-4194
- *IBM Tivoli Composite Application Manager Family Installation, Configuration, and Basic Usage*, SG24-7151
- *Implementing IBM Director 5.20*, SG24-6188
- *Introduction to Workload Partition Management in IBM AIX Version 6.1*, SG24-7431
- *NIM from A to Z in AIX 5L*, SG24-7296
- *PowerVM Live Partition Mobility on IBM System p*, SG24-7460
- *PowerVM Virtualization on IBM System p: Introduction and Configuration Fourth Edition*, SG24-7940
- *PowerVM Virtualization on IBM System p Managing and Monitoring*, SG24-7590
- *Quantifying the Benefit of EWLM in a System p Environment*, REDP-4394
- *Virtualization and Clustering Best Practices Using IBM System p Servers*, SG24-7349
- *Virtualizing an Infrastructure with System p and Linux*, SG24-7499

Other publications

These publications are also relevant as further information sources:

- Gamma, et al., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994, ISBN 0201633612

Online resources

These Web sites are also relevant as further information sources:

- AIX info center

  http://publib.boulder.ibm.com/infocenter/systems/scope/aix/index.jsp

- AIX related Web pages

  http://www.ibm.com/systems/p/os/aix/certifications/index.html
  http://www.ibm.com/systems/p/os/aix/sysmgmt/me/index.html

- bash rpm package from the Linux toolbox

  http://www.ibm.com/systems/p/os/aix/linux/toolbox/download.html

- DB2 installation guide

  http://publib.boulder.ibm.com/infocenter/db2luw/v9/topic/com.ibm.db2.udb.uprun.doc/doc/t0008875.htm

- IBM Linux on POWER Web pages

  http://www.ibm.com/systems/linux/power

- *IBM POWER6 Processor-based Systems: Designed for Availability*, found at:

  http://www-03.ibm.com/systems/p/hardware/whitepapers/power6_availability.html

- IBM SOA developer works

  http://www.ibm.com/developerworks/soa
  http://www.ibm.com/developerworks/webservices/standards/
  http://www.ibm.com/developerworks/webservices/library/ws-antipatterns/

- *IBM System p5: A Highly Available Design for Business-Critical Applications*, found at:

  http://www.ibm.com/servers/eserver/pseries/library/wp_lit.html

- IBM System p POWER6 pages

  http://www.ibm.com/systems/p/hardware/whitepapers/power6_availability.html
  http://www-03.ibm.com/systems/p/advantages/cod/contact/vpd_form.html
  http://www14.software.ibm.com/webapp/set2/sas/f/vios/documentation/datasheet.html
  http://www14.software.ibm.com/webapp/set2/sas/f/hmc/home.html

- Installation of WebSphere MQ instructions

  http://www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=GC34647800

- ITCAM for Response Time Tracking page

  http://www.ibm.com/software/tivoli/products/composite-application-mgr-rtt/

- ITCAM for SOA page

  http://www.ibm.com/software/tivoli/products/composite-application-mgr-soa/

- ITCAM for WebSphere page

  http://www.ibm.com/software/tivoli/products/composite-application-mgrwebsphere/

- Latest fixes for WebSphere MQ

  http://www-1.ibm.com/support/docview.wss?rs=171&uid=swg21254675

- launchpad_v2.0_fixpack18.zip attachment file

  http://www-1.ibm.com/support/docview.wss?uid=swg21288640&aid=1

- Object Management Group SOA page

  http://soa.omg.org/

- Pluggable Authentication Method page

  http://www.kernel.org/pub/linux/libs/pam/index.html

- Red Hat Linux Web pages

  http://www.redhat.com

- Service and productivity tools for Linux on POWER

  https://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/home.html

- SOA Governance page

  http://www.ibm.com/software/solutions/soa/gov/method/
  http://www.ibm.com/pod/pod

- SUSE Linux Web pages

  http://www.novell.com/linux
  http://www.novell.com/products/server/

- Steel Eye LifeKeeper

  http://www.steeleye.com/
  http://www.beowulf.org
  http://www.myri.com/

- System automation on Linux

  http://www.ibm.com/software/tivoli/products/sys-auto-linux/

- TPC performance benchmarking result

  http://www.tpc.org/tpcc/results/tpcc_perf_results.asp

- Web Services integration page

  http://www.ws-i.org

- WebSphere Service Registry and Repository installation

  http://publib.boulder.ibm.com/infocenter/sr/v6r1/topic/com.ibm.sr.doc/twsr_installn17_2.html

- WebSphere V6.1 Fix Pack 13

  http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg24012718

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

abstraction 11
access control list
  See ACL
access services 57
ACL 195
antipatterns 95
Application Response Measurement
  See ARM
Application Server 190
Application Server Facilities
  See ASF
architecture implementation 53
ARM 121
ASF 28
autonomy 12
availability 116

B

binformation services 57
BPEL 44
BPM 33
business application services 56
business driven entry points 68
business innovation 59
Business Process Execution Language
  See BPEL
Business Process Management
  See BPM

C

Capacity BackUp 112
Capacity on Demand (CoD) 154
Capacity Upgrade on Demand 111
CAPP 138
capped 110
central processing unit
  See CPU
chmod command 226
commands
  chmod 226
  cp 192
  install 224
  install.sh 192
  installall.sh 196
  launchpad.sh 190
  manageprofiles.sh 225
  mksysb 153
  mkwpar 223
  mqsicreateaclentry 195
  probevue 140
  slibclean 226
  smitty 222
  typeset 191
Common Object Request Broker Architecture
  See CORBA
composability 12
Composite Application Manager 166
concurrent maintenance 125
Controlled Access Protection Profile
  See CAPP
CORBA 8
cp command 192
CPU 120

D

DAC 131
data-oriented programming
  See DOP
DataPower Integration Appliance 30
DB2 Universal Database Enterprise Server Edition V9.1 193
development services 59
discoverability 12
Discretionary Access Control
  See DAC
DLPAR 108
DNS 10
Domain Name System
  See DNS
DOP 31
dynamic logical partitioning
  See DLPAR

E

EAL4+ 130
ECN 198
EFS 134
EJB 44
electronic communication network
See ECN
Electronic Service Agent 124
Encrypting File System
See EFS
Energy Management 122
Enhanced Journaled File System
See JFS2
Enterprise Java Bean
See EJB
enterprise resource planning
See ERP
Enterprise Service Bus
See ESB
Enterprise Workload Manager 120
ERP 29
ESB 22, 27, 58, 209
Evaluation Assurance Level 4+
See EAL4+

F

FFDC 141
First Failure Data Capture
See FFDC

H

HACMP 147
Hardware Management Console
See HMC
High Availability Cluster Multiprocessing
See HACMP
HMC 115, 119
Hypervisor 106

I

I/O virtualization 112
IBM DB2 Universal Database Enterprise Server Edition 193
IBM Director 123, 165
IBM PowerVM Lx86 105
IBM Tivoli Composite Application Manager
See ITCAM
IBM Tivoli Monitoring
See ITM
IBM Tivoli System Automation for Multiplatform 158
Image repository 153
Infrastructural antipatterns 95
infrastructure management 83
infrastructure requirements 74
Infrastructure services 60
infrastructure virtualization 85
install command 224
install.sh command 192
installall.sh command 196
Integrated Virtualization Manager
See IVM
interaction services 55
iPDU support 123
IT Service Management
See ITSM
IT service management 60
ITCAM 47, 166
ITM 84, 166, 214
ITSM 68
IVM 120, 178

J

Java Database Connectivity
See JDBC
Java Messaging Services
See JMS
JDBC 29
JFS2 134
JMS 21

L

Labeled Security Protection Profile
See LSPP
launchpad.sh command 190
LDAP 29, 136
Lightweight Directory Access Protocol
See LDAP
Live Partition Mobility 106
logical architecture 54
logical partitioning 108
loose coupling 10, 25
LPAR 154
LSPP 130, 138, 160

M

MAC 132
management features 119
manageprofiles.sh command 225
Mandatory Access Control
  See MAC
master image 152
mediation considerations 78
mediations 19
memory virtualization 112
Message Broker 28, 194, 209
misconceptions 17
mksysb command 153
mkwpar command 223
MLS 132
mobility 150
Monitoring 166, 214
MQ 193
mqsicreateaclentry command 195
multi-level security
  See MLS

N

National Security Agency
  See NSA
Network Installation Manager
  See NIM
NFR 51
NIM 145, 152, 164
non-functional requirements
  See NFR
NSA 139

O

Object Management Group
  See OMG
OGSA 74
OMG 6
On/Off Capacity on Demand 111
Open Grid Services Architecture
  See OGSA
optimization services 59

P

partner services 58
performance considerations 82
Power Capping 123
Power Savings Mode 123
Power Systems 3
Power Trending 123
probevue command 140
process services 56
Processor virtualization 108
product
  Application Server 190
  Composite Application Manager 166
  DataPower Integration Appliance 30
  ESB 27, 209
  Message Broker 28, 194, 209
  Monitoring 166, 214
  MQ 193
  Services Registry and Repository 196
  System Automation for Multiplatform 158
  Universal Database Enterprise Server Edition 193

R

RAS 116
RBACPP 130
RBPP 138
RDS 121
Redbooks Web site 240
  Contact us xiv
reference architecture layers 48
registry 19
registry considerations 78
reliability 116
reliability, availability, serviceability
  See RAS
Reliable Scalable Cluster Technology
  See RSCT
Remote Method Invocation
  See RMI
Resource Dependency Service
  See RDS
Resource Monitoring Control
  See RMC
return of investment
  See ROI
reusability 11
RMC 144
RMI 8
ROI 189
Role Based Access Control Protection Profile
  See RBACPP

Role Based Protection Profile
See RBPP
RSCT 144

S

SAN 86, 113
SEA 114
Second Failure Data Capture
See SFDC
security considerations 80
service autonomy 12
service contract 9
service creation 63
Service Focal Point
See SFP
service level agreement
See SLA
service-oriented architecture
See SOA
services 8
abstraction 11
composability 12
discoverability 12
reusability 11
statelessness 12
Services Registry and Repository 196
SFDC 141
SFP 188
SGMM 48
Shared Ethernet Adapter
See SEA
Simple Object Access Protocol
See SOAP
SLA 9, 120
slibclean command 226
smitty command 222
SOA 2, 5, 30, 69, 189, 196
basics 2
benefits 16
definition 12
design 67
entry points 68
evolution 6
governance 67
infrastructure selection 73
introduction 5
life cycle 40
mediation 19
misconceptions 17
reference architecture 48
registry 19
scenarios 63
security and management 67
solution stack 48
standards 19–20
use cases 71
SOA entry points
Connectivity 70
Information 69
IT driven entry points 70
People 68
Process 69
Reuse 70
SOA Governance and Management Method
See SGMM
SOA infrastructure 87
key elements 77
SOA scenarios 61
SOA scnearios
Information as service 67
Interaction and collaboration services 66
service connectivity 65
SOAP 21
SPT 178
statelessness 12
SteelEye LifeKeeper 160
Storage Area Network
See SAN
System Automation for Multiplatform 158
System plan 153, 170
System Planning Tool
See SPT
system template 152
Systems Provisioning 122

T

Thermal Trending 123
Tivoli Composite Application Manager 166
Tivoli Monitoring 166, 214
Tivoli System Automation for Multiplatform 158
Trial Capacity on Demand 112
Trusted Signature Database
See TSD
TSD 137
typeset command 191

U

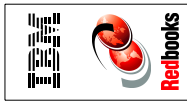
UDDI 35
uncapped 110
Universal Database Enterprise Server Edition 193
Universal Discovery, Description, Integration
See UDDI

V

VIOS 162
Virtual Ethernet 113
virtual I/O 150
Virtual I/O Server 178
See VIOS
Virtual LAN
See VLAN
virtual networking 149
Virtual SCSI 114
virtual server 153, 172
virtualization 104
Virtualization Manager 175
Virtualization Manager extension 124
VLAN 113

W

Web Services Description Language
See WSDL
Web Services Distributed Management
See WSDM
Web Services for Remote Portlets
See WSRP
WebSphere Application Server 190
WebSphere DataPower Integration Appliance 30
WebSphere Enterprise Service Bus 27, 209
WebSphere Message Broker 28, 194, 209
WebSphere MQ 193
WebSphere Service Registry and Repository
See WSRR
WLM 229
Workload Manager
See WLM
workload partition
See WPAR
WPAR 107, 145, 153
WSDL 21, 28
WSDM 74
WSRP 50
WSRR 35, 196, 209
business goal 36
role 34



Power Systems and SOA Synergy

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Power Systems and SOA Synergy

Explains partitioning and virtualization technologies

Discusses SOA implementation on System p

Covers the AIX and Linux platforms

This IBM Redbooks publication gives a broad understanding of the synergy between service-oriented architecture (SOA)-based applications and Power Systems servers. The popularity and reach of SOA-based applications has grown exponentially in recent years. Enterprises are relying more on SOA-based applications for their operation. As a mission critical system, it is critical that the application be supported by an adequately planned infrastructure.

IBM Power Systems have been leading players in the server industry for decades. Power Systems provide great performance while delivering reliability and flexibility to the infrastructure. Given the advent of SOA-based applications, this book aims to demonstrate what benefits a SOA-based application can get from a Power Systems infrastructure and how Power Systems support a SOA-based application.

The book is intended as a guide for a Power Systems specialist to understand the SOA environment and for a SOA specialist to understand the facilities available for Power Systems supporting SOA-based applications.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7607-00 ISBN 0738485357