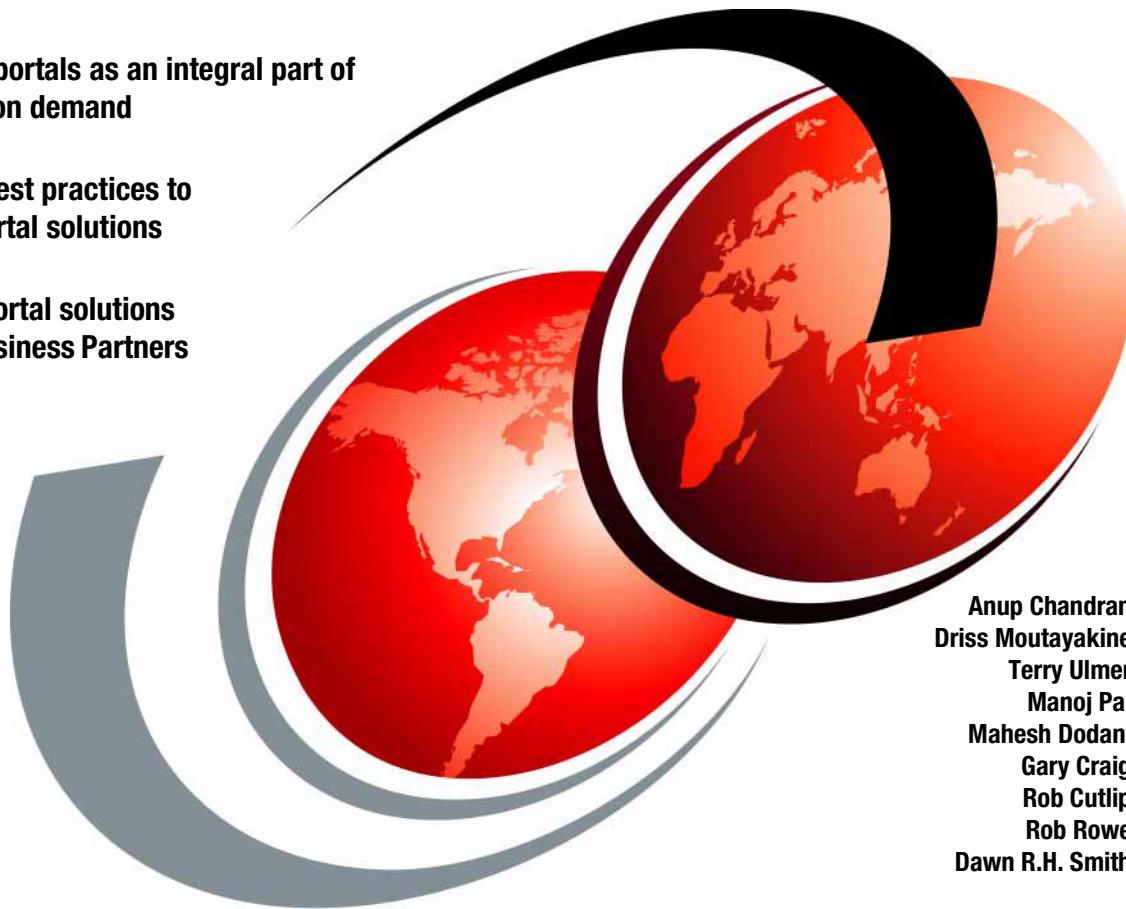


Architecting Portal Solutions

Positioning portals as an integral part of
e-business on demand

Using IBM best practices to
architect portal solutions

Extending portal solutions
with IBM Business Partners



Redbooks



International Technical Support Organization

Architecting Portal Solutions

December 2003

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

First Edition (December 2003)

This edition applies to Version 5 of IBM WebSphere Portal, Application Server, Commerce, and Studio.

© Copyright International Business Machines Corporation 2003. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
Prefacexi
The team that wrote this redbook.....	xii
Become a published author	xv
Comments welcome.....	xv
Part 1. A best practice guide to architecting portal solutions	1
Chapter 1. Overview	3
1.1 On demand	5
1.2 Portals	12
1.3 Patterns.....	28
Chapter 2. Portal solutions.....	39
2.1 Introduction	39
2.1.1 What is a portal?.....	40
2.1.2 Types of portals.....	41
2.1.3 A brief history of portals	42
2.1.4 Functional aspects of a portal.....	43
2.1.5 Technical aspects of a portal	45
2.2 Architecting portal solutions	49
2.2.1 Deriving value from portal solutions	50
2.2.2 Portal solution best practices	55
2.2.3 Steps for building a successful portal solution	57
2.3 Portals and e-business on demand.....	58
2.3.1 What is an e-business on demand?	58
2.3.2 Building the on demand operating environment	59
2.3.3 The evolution of e-business on demand	59
2.3.4 How portals can lead to an e-business on demand	61
2.3.5 Portals in an on demand operating environment	62
2.4 Summary.....	63
Chapter 3. A recommended approach to architecting portal solutions ..	65
3.1 Plan	77
3.2 Execute	80
3.2.1 Step 1: Develop solution requirements	84
3.2.2 Step 2: Develop the logical design	97

3.2.3 Step 3: Develop the physical solution	142
3.3 Implement	165
Chapter 4. Architecting On Demand Workplace solutions	167
4.1 Retail case study introduction	168
4.1.1 Company background.....	168
4.1.2 Core functions.....	168
4.2 Plan	169
4.2.1 Business requirements	169
4.2.2 IT requirements.....	170
4.3 Execute (develop solution requirements)	170
4.4 Execute (develop the logical design).....	177
4.4.1 Transition to AOD (logical nodes).....	185
4.5 Execute (develop the physical design)	190
4.6 Reflection on ODW	195
4.6.1 ODW generic use cases	195
Part 2. Extending portal solutions with IBM Business Partners	199
Chapter 5. Galaxia and the On Demand Workplace.....	201
5.1 Galaxia solutions for the industrial sector	202
5.1.1 Galaxia and ePCM	202
5.1.2 Industry overview	202
5.1.3 Vehicle development process.....	204
5.1.4 Key business drivers.....	208
5.1.5 Key infrastructure and IT drivers.....	210
5.2 The solution.....	212
5.2.1 Requirement summary	212
5.2.2 Generic use cases for the solution	213
5.2.3 Solution Overview Diagram and identification of patterns.....	216
5.2.4 Identification of Application patterns	218
5.2.5 Applying the delta Runtime patterns to the Portal composite runtime pattern.....	223
5.2.6 Product mapping.....	226
5.2.7 On Demand Workplace.....	229
5.3 The Galaxia demo scenario	230
5.3.1 Steps in a typical ECR process.....	230
5.3.2 Product mapping for the scenario	232
5.3.3 Interactions between various applications in the scenario	235
5.3.4 ECR process scenario	236
5.4 Galaxia ePCM	242
5.4.1 ePCM product overview	242
5.4.2 Templates for engineering processes	245
5.4.3 Templates for the Work Management and Task Allocation process	246

5.4.4 Template for order processing	247
5.4.5 Workflow federation	248
5.4.6 Other ePCM features	248
5.4.7 ePCM architecture	252
5.5 Summary	254
Part 3. Appendixes	257
Appendix A. e-business on demand	259
A.1 e-business on demand: A developer's roadmap.....	260
A.2 Resource center: e-business on demand	271
A.2.1 Open standards	272
A.2.2 Integration.....	273
A.2.3 Virtualization.....	321
A.2.4 Automation	346
Appendix B. Using WebSphere Portal.....	383
B.1 WebSphere Portal architecture.....	384
B.1.1 Portlets	386
B.1.2 Content and search	388
B.1.3 Security	391
B.1.4 Personalizing the portal	394
B.1.5 Administration.....	398
B.1.6 Collaboration.....	401
B.1.7 Related products.....	402
B.1.8 Application integration	404
B.1.9 Mobile portals	406
B.2 Developer information roadmap	408
B.2.1 Basic skills	408
B.2.2 Installation/configuration.....	409
B.2.3 Migration	409
B.2.4 Developing portlets.....	409
B.3 What's new with Version 5.0.....	416
B.3.1 New features.....	416
B.3.2 Installation improvements.....	417
B.3.3 Improvements for administering and designing WebSphere Portal	418
B.3.4 Removed features	420
B.3.5 Contents	420
B.3.6 Available with WebSphere Portal Extend	423
B.3.7 Supplementary products.....	426
Related publications	429
IBM Redbooks	429
Other publications	430

Online resources	430
How to get IBM Redbooks	434
Help from IBM	434
Abbreviations and acronyms	435
Glossary	439
Index	447

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 illustrates 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. 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 IBM's application programming interfaces.

Trademarks

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

@server™	eServer™	Rational Rose®
alphaWorks®	ibm.com®	Rational Unified Process®
Approach®	IBM®	Rational®
AS/400®	Illustra™	Redbooks (logo)  ™
Balance®	IMS™	Redbooks™
CICS®	Informix®	RUP®
Cloudscape™	iNotes™	S/390®
CrossWorlds®	Intelligent Miner™	Sametime®
DataBlade™	iSeries™	ServeRAID™
DataJoiner®	Lotus Discovery Server™	Sysplex Timer®
DataPropagator™	Lotus Notes®	ThinkPad®
DB2 Extenders™	Lotus®	Tivoli Enterprise Console®
DB2 Universal Database™	MQSeries®	Tivoli Enterprise™
DB2®	Netfinity®	Tivoli®
developerWorks®	NetView®	TotalStorage®
Domino.Doc®	Notes®	@server™
Domino®	PartnerWorld®	WebSphere®
Dynamic Workplaces™	Perform™	XDE™
e-business on demand™	QBIC®	xSeries®
Electronic Service Agent™	QuickPlace®	zSeries®

The following terms are trademarks of other companies:

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

ENOVIA, ENOVIA PPR, ENOVIA VPM, 3d com, and CATIA V4 and V5 are registered trademarks of Dassault Systèmes.

Teamcenter Enterprise is a registered trademark of UGS PLM Solutions, anEDS company.

SAP and SAP R/3 are registered trademarks of SAP.

Oracle is a registered trademark of Oracle Corporation.

Solaris and JMS are registered trademarks of Sun Microsystems.

HP-UX is a registered trademark of Hewlett-Packard.

MS Windows, Windows NT, and Windows 2000 are registered trademarks of Microsoft.

ePCM is a registered trademark of Galaxia.

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

Preface

This IBM Redbook is focused on architecting and building WebSphere Portal Server-based Dynamic Workplaces. It addresses the needs of SWG Architects, Business Partners, and customers for building skills in architecting solutions by identifying themes that can be applied across multiple SWG Industry Solutions. This includes using e-business infrastructure solutions maps that apply across multiple industries: for example, the On Demand Workplace solution, the e-business integration stack from SWG that provides capabilities for integrating the enterprise based on both business and technology drivers, and the IBM® Patterns for e-business from SWG that provide the basis for understanding the best practices for supporting e-business integration. This redbook provides real examples of industry-specific Dynamic Workplace solutions from leading Solution Developers and Systems Integrators working within the IBM on demand strategy. It also includes information on how to use IBM software capabilities to fulfill customer requirements using WebSphere, Lotus, Rational, Tivoli, and Data Management products, how portals are used in various industries, and how this investment can lead a customer towards becoming an e-business on demand.

Over the last four years, many top architects in this community have successfully exploited the Patterns for e-business and TeAMethod because they intuitively understood the way to fit these together. We now recognize that we can help all architects by making the linkage more explicit—and in doing so we have realized we can add one or two more work products to the Patterns deliverables to increase everyone's asset reuse when building a proposal using TeAMethod.

We also plan to propagate these ideas to your colleagues in IBM Global Services so that Global Services method practitioners also get a more explicit linkage to the Patterns or other reusable assets.

Please continue to give us feedback on your successes and challenges in applying this approach. The Software Architects Skills Board is committed to continuing to improve the combination of TeAMethod and the Patterns for e-business.

Regards,

Jonathan Adams, author, Patterns for e-business, and Distinguished Engineer, SWG Technical Strategy

Note: The use of “to architect” and “architecting” as verbs are emerging in the field to address the limitations imposed by the verbs “design” and “build” to capture what architects do. In this redbook, the word “architecting” is used to describe all of the activities that architects perform to develop a solution, including understanding the customers’ environment, understanding and defining the requirements, analyzing the system, designing the solution, and managing the solution development process.

The team that wrote this redbook

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

Mahesh Dodani is an e-business architect with IBM Software Group. His primary interests are in enabling individuals and organizations to tackle complex e-business industry solutions. He can be reached at dodani@us.ibm.com.

Gary Craig is president of Superlative Software Solutions, Inc., where his roles include senior consultant specializing in technical enablement, training, and mentoring for J2EE development and deployment as well as IBM WebSphere® brand products and technology. Gary holds an MS and Ph.D. in Electrical Engineering from the University of Wisconsin at Madison and taught at Syracuse University. He is a senior member of the IEEE and co-authored *Enterprise Java Programming with IBM WebSphere*¹ and *An Introduction to Web Application Development with IBM WebSphere Studio*.² Dr. Craig can be reached at glcraig@us.ibm.com.

Rob Cutlip is a software and solutions architect with the IBM Software Group based in Research Triangle Park, NC. He has spent the last seven years building integrated industry solutions. An author and inventor, Rob has 15 years of experience in both technical and managerial positions with Fortune 500 companies. He is co-author with John Medicke of *Integrated Solutions with DB2*.³

Rob Rowe is a senior IT Specialist for IBM Global Services, currently focusing on cross-brand industry solutions demonstrations and technical enablement. In addition to software development of numerous applications and demonstrations, he has created IBM courses consisting of both lecture and laboratory exercises and has written certification exams for WebSphere products. He is responsible for the IBM ServeRAID™ Raid Array Controller running on UNIX and Linux platforms. Prior to joining IBM, he held many different positions, including a

¹ Addison-Wesley, October 2003 (ISBN 0-321185-79-X)

² MC Press, October 2003 (ISBN 1-931182-11-6)

³ Addison-Wesley, November 2003 (ISBN 0-201754-85-1)

technical instructor and course developer, a software developer for a large computer corporation, a consultant focusing on software development and writing technical material, and a manager of a development, test, and qualification organization dealing with storage products running on numerous UNIX platforms. He resides in Raleigh, NC.

Dawn R.H. Smith is an IT Specialist within IBM Global Services specializing in developing WebSphere applications. She is currently responsible for creating portable, cross-brand solutions for various industries. Through collaboration with different organizations (ODSC, GeSC, SWG, etc.), these solutions are used to demonstrate IBM's ability to solve real-world business problems. These demonstrations are used for sales and technical enablement. Her prior experience spans a variety of roles and technologies. She played a critical role in the development and education world-wide for IBM's Project Management certification application. She has also contributed to SWG's Architecting Portal Solutions course, as well as delivering numerous development solutions utilizing WebSphere.

Anup Chandran is an AIM Specialist at IBM Business Partner Galaxia. He received his Bachelors in Engineering from Mysore University, India in 1995. He has a Diploma in Business Management from Goa Institute of Management, India. His career began as a software developer and progressed into various roles such as coordinator, technical sales support, and analyst. He has been with Galaxia for the last two years, primarily involved in developing enterprise workflow solutions using IBM products for various automotive and aerospace companies. His expertise lies in business process analysis and modeling. You can reach him at acn@galaxia.ca.

Driss Moutayakine has over eight years' experience in the aerospace and automotive industry, having worked for Daimler Chrysler, Dassault Systèmes, EDS, ENOVIA, Lockheed Martin, and IBM. He has made presentations to many customer forums in Europe, Asia, North America, and Africa. Driss holds a bachelor's degree in Computer Engineering from the Polytechnique School of Montreal. For the past two years, he has managed ePCM product development and strategy for Galaxia. He is IBM-certified and has received project management training from the Project Management Institute.

Terry Ulmer is an IT Architect with IBM Global Services Americas. He is a 22-year IBM veteran specializing in enterprise-level solutions using Web technologies. Recently, he architected and led development of the Integrated Operator Interface, a strategic portal solution currently in use at IBM's Microelectronics Semiconductor Fabricators worldwide, including the new 300 mm fab in East Fishkill, NY. Currently, Terry is technical lead for IBM Electronic Service Agent™ for xSeries® and Netfinity®. He can be reached at tulmer@us.ibm.com.

Manoj Pal is a certified Enterprise Architect and holds varied technical certifications including OOAD/UML, WebSphere Application Server, EJB, Java, and DB2 Specialist as Administrator/Developer/User. He is currently working in WebSphere Business Component development and with the San Francisco Java Framework team in the IBM Software Lab of India. His focus in recent years has been Web services, EJB Business Components and Frameworks, EAI, and Portals. Manoj has more than ten years of experience, including working in Process Industry, stand-alone to *n*-tier distributed business applications, 16-bit DOS to AS/400, and VB to J2EE. Manoj can be reached at pkmanoj@in.ibm.com or manojpal@lycos.com.

Editorial staff

Buck Stearns of ITSO Raleigh served as managing editor.

Cecilia Bardy of ITSO Raleigh served as executive editor.

Linda Robinson of ITSO Raleigh served as graphics editor.

Reviewers

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

Gary McGibbon, Senior Manager, ISV and Developer Relations, IBM Software Group, RTP, NC

Dave Coffman, Technology Manager, ISV and Developer Relations, IBM Software Group, RTP, NC

Greg Labows, Technology Manager, ISV and Developer Relations, IBM Software Group, RTP, NC

Bernie Clark, ENOVIA Product Marketing, Industrial Sector EMEA, IBM UK, Glasgow

Larry Carvalho, Market Segment Manager, Software Solutions for Automotive, Industry Market Management, IBM Software Group, Cincinnati, OH

Jonathan Adams, Distinguished Engineer, IBM Software Group Technical Strategy, Cheshire, UK

Sew Bun Foong, Manager, IBM ASEAN Consulting IT Architects, Singapore

Kanumury Radhesh, Software IT Architect, IBM India

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll 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 Redbooks™ to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

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

ibm.com/redbooks

- ▶ Send your comments in an Internet note to:

redbook@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 662
P.O. Box 12195
Research Triangle Park, NC 27709-2195



Part 1

A best practice guide to architecting portal solutions

This redbook focuses on bringing together the various efforts in engineering complex e-business solutions used within IBM. To be able to describe these efforts effectively, we have chosen to concentrate on people integration (or *portal*) solutions, which allow a company to provide a personalized, integrated, interactive view of its business by giving customers, suppliers, partners, and employees anytime/anywhere access to information, transactions, and

know-how. Key integration requirements include role-based, preference-based, and device-based transformation of business transactions and information.

Chapter 1, “Overview” on page 3 provides details on some of the main topics that provide the foundation for this redbook—on demand computing, portals, and patterns. The approach taken in explaining these concepts is through a set of questions and answers that focuses the discussion on the areas pertinent to the redbook.

Chapter 2, “Portal solutions” on page 39 provides a detailed look at portal solutions and sets up the class of problems that we will use to showcase the design of complex solutions using best practices.

Chapter 3, “A recommended approach to architecting portal solutions” on page 65 presents our recommended approach to architecting portal solutions by infusing the use of large assets into an industrial-strength application development methodology. The large assets that can be leveraged include IBM Patterns for e-business, reference architectures, industry solutions along with their reference implementations, and frameworks. The best practices from field experiences are incorporated as guidelines for designing and building the solution.

Chapter 4, “Architecting On Demand Workplace solutions” on page 167 applies the recommended approach to a class of portal solutions targeted at establishing an e-workplace for employees of an enterprise.

Part 2, “Extending portal solutions with IBM Business Partners” on page 199 then showcases how business partner Galaxia extended the approach to architecting portal solutions articulated in Part 1 to incorporate its solutions and offerings in conjunction with IBM. This part provides a blueprint for other business partners who want to include their solutions effectively with IBM and also to leverage the reusable assets and best practices in building these solutions.



Overview

The complexity of computer applications and systems continues to grow at an exponential rate. In a very short period of time, we have evolved from single applications addressing a specific functionality or set of requirements on a dedicated mainframe to an integrated set of applications representing an enterprise e-business. What is an *e-business*? This is a catch-all term for identifying an enterprise that has transformed its business by:

- ▶ Leveraging Web technologies to reengineer business processes
- ▶ Enhancing communications and lowering organizational boundaries with its:
 - Customers and shareholders (across the internet)
 - Employees and stakeholders (across the corporate intranet)
 - Vendors, suppliers, and partners (across the extranet)

This e-business transformation is evolutionary and complex. The typical evolution for an enterprise is to move from:

- ▶ Providing access to enterprise applications and information
- ▶ Through integration of systems and applications in the enterprise
- ▶ To an adaptive enterprise that can optimize operations and dynamically adapt to the needs of all its constituents

We have already embarked on the next wave of change and complexity in e-business and entered the *on demand* era.

Definition: An *on demand business* is an enterprise whose business processes—integrated end-to-end across the company and with key partners, suppliers, and customers—can respond with flexibility and speed to any customer demand, market opportunity, or external threat.

In this *on demand* world, companies will move beyond simply integrating their processes to actually being able to sense and respond to fluctuating market conditions and provide products and services to customers on their terms—*on demand*. Companies will be able to acquire business functions or IT infrastructure over the Internet as they need them and only pay for what they use. Companies can quickly increase or decrease their requirements as their markets change. IBM has developed products and services that meet the business and infrastructure requirements for enabling business on demand and can help customers begin building their on demand capabilities today.

The methods, approaches, products, and tools used to design, build, and implement such on demand systems must also evolve at a rapid pace to keep up with ever-growing complexity. Several work efforts on reference architectures, patterns, frameworks, and open standards have all contributed to supporting the ability to handle this complexity. The common thread of these work efforts is that they capture best practices and proven experiences into customizable, reusable assets. Naturally, the larger the asset, the more useful it is in helping drive the entire solution and the more difficult it is to reuse. To be effective, we must facilitate the reuse of these large assets in the context of familiar methodologies and work products that architects use to design their solutions.

This redbook focuses on bringing together various efforts used within IBM to engineer complex solutions. To be able to describe these efforts effectively, we have chosen to focus on people integration (or *portal*) solutions.

Definition: *People integration solutions* allow a company to provide a personalized, integrated, interactive view of its business by giving customers, suppliers, partners, and employees anytime/anywhere access to information, transactions, and know-how. Key integration requirements include role-based, preference-based, and device-based transformation of business transactions and information.

Chapter 2, “Portal solutions” on page 39 provides a detailed look at portal solutions and sets up the class of problems that we will use to showcase the design of complex solutions using best practices. Chapter 3, “A recommended approach to architecting portal solutions” on page 65 presents the main recommended approach to architecting portal solutions by infusing the use of large assets into an industrial-strength application development methodology.

The large assets that can be leveraged include IBM Patterns for e-business, reference architectures, industry solutions along with their reference implementations, and frameworks. The best practices from field experiences are incorporated as guidelines for designing and building the solution. Chapter 4, “Architecting On Demand Workplace solutions” on page 167 applies the recommended approach to a class of portal solutions targeted at establishing an e-workplace for employees of an enterprise.

The rest of Chapter 1 provides details on some of the main topics that provide the foundation for this redbook—on demand computing, portals, and patterns. The approach we have taken is to explain these concepts through a set of questions and answers that focuses on areas pertinent to the redbook. Readers who are familiar with these topics can skip through these sections.

1.1 On demand

1. What does IBM mean by *on demand*?

A totally new kind of transformation—or, more specifically, new levels of integration—of processes and applications inside the business, of suppliers and distributors at either end of the business, of customers outside the enterprise, of employees inside it.

e-business on demand is the name we use to describe a new movement in business and computing, one that is changing the way technology is deployed and used in business. The dynamics of the market are changing, and that requires a new way of thinking about business processes and the information technology infrastructure that supports them. We are entering the next phase of e-business, where companies move beyond simply integrating their various processes to a world in which they need to be able to sense and respond to fluctuating market conditions in real time. IBM is leading the way toward the on demand world, and has the business insight and technology expertise to help our customers become on demand businesses.

Definition: *On demand business*—an enterprise whose business processes—integrated end-to-end across the company and with key partners, suppliers, and customers—can respond with flexibility and speed to any customer demand, market opportunity, or threat.

2. What is *e-business on demand™*?

See “[Just what is e-business on demand?](#)” on page 261.

3. What does an on demand business look like?

An *on demand business* is an enterprise whose business processes—integrated end-to-end across the company and with key partners, suppliers, and customers—can respond with agility and speed to any customer demand, market opportunity, or external threat. An on demand business:

- a. Is **responsive**—Responding almost intuitively to dynamic, unpredictable changes in demand, supply, pricing, labor, competitors' moves, capital markets, and the needs of all its constituencies—customers, partners, suppliers, and employees.
- b. Uses **variable cost structures** and adapts processes flexibly. This flexibility will enable it to reduce risk and to do business at high levels of productivity, cost control, capital efficiency, and financial predictability.
- c. Is **focused on its core competencies**—its differentiating tasks and assets—while tightly integrated strategic partners manage selected tasks (everything from manufacturing, logistics, and fulfillment, to HR (Human Resources) and financial operations).
- d. Is **resilient** enough to manage changes and threats—from computer viruses to earthquakes to spikes in usage—with consistent availability and security.

An on demand business enables its leaders to see and manage the company as an integrated whole, even if important parts of its business are handled by others.

Business characteristics of an on demand environment

See “[Business characteristics of an on demand environment](#)” on page 263.

4. What kind of technology infrastructure is needed to support on demand capabilities?

To become on demand businesses, companies must deploy IT (Information Technology) to create an on demand operating environment. This has four essential characteristics:

- a. It is **integrated**—Systems are seamlessly linked across the enterprise and across its entire range of customers, partners, and suppliers.
- b. It uses **open standards**—So different systems can work together and link with devices and applications across organizational and geographic boundaries.
- c. It is **virtualized**—To make the best use of technology resources and minimize complexity for users, it uses grids to make the collective power of the computing resources in the grid available to anyone there who needs them.

- d. It has **self-healing, autonomic capabilities**—so it can respond automatically and work around problems, security threats, and system failures.

On demand businesses need reliable, scalable, and secure infrastructures that will be used by customers, employees, suppliers, and contractors. These infrastructures will be the foundation of all business processes—supply chains, HR systems, customer fulfillment, financial applications—and will handle billions of dollars worth of transactions and vital communications.

Businesses must meet usage demands regardless of technology or affordability issues. These businesses will need to upgrade and improve continuously to become more flexible and resilient. This new environment will create tremendous opportunity for IBM software, hardware, and services, as well as for new applications provided by IBM's market-leading Solution Developer partners.

Employees will need pervasive access to interact with information in whatever mode is appropriate for the context, whether text, voice, or some combination of modes—desk-bound or mobile.

The on demand operating environment has four essential characteristics—*integrated, open, virtualized, and autonomic*.

Technical characteristics of an on demand environment

See "[Technical characteristics of an on demand environment](#)" on page 264.

5. Why should our customers want to become on demand businesses?

Customers of every size and in every industry are looking for ways to make their businesses more resilient and agile in the face of change and uncertainty. The reality is that in today's environment, they must be able to make faster and better decisions, reduce risk, leverage their core competencies, and increase their return on investment. Becoming an on demand business is about changing the way you operate as well as reducing costs, serving your customers better, reducing risks, and improving speed and agility in the marketplace. A business can only meet these needs if it has on demand capabilities, and those that develop them early will gain considerable competitive advantages. IBM has the business expertise and technology resources and experience to help our customers become on demand businesses.

6. Why now?

The pace of business has increased, the networked world has made every company a global competitor, customer expectations are rising, and information technology has only begun to realize its promise to transform business. Companies that begin building their on demand capabilities now have the opportunity to gain significant advantages over their competitors.

7. What are the phases of e-business on demand?

See “[Phases of e-business on demand](#)” on page 262.

8. How is e-business on demand different from the e-business IBM has been talking about for years?

The early phase of e-business was about communication—having a presence on the Internet, communicating inside and outside your company, and making some information available online. The second phase was about transactions—being able to perform basic transactions online. This next phase is a qualitative shift, going beyond changing a standalone process to seamlessly integrating business processes across the entire range of business activity, from procurement to finance to logistics to customer-facing systems—from customers to partners to suppliers to employees.

Previously, the words, “on demand” were often used in reference to “e-sourcing,” or utility-like computing. While utility-like computing can be one of the technology resources that help businesses develop on demand capabilities, this is a much broader idea about the transformation of both business and technology. Companies in the on demand world will have the capacity to sense and respond to fluctuating market conditions in real time and to provide products and services to customers on demand. IBM has been developing services, technologies, and integrated solutions that support on demand capabilities, and we are focused on helping our customers become on demand businesses.

9. What are the key competitive advantages IBM has to help customers become on demand businesses?

IBM is defining and leading the next big change in business and computing—setting the agenda for the industry. We have been preparing for this for some time, and our products and services already are delivering on demand capabilities and benefits to our customers.

No other company has the breadth and depth of our business and technology experience. We have the experience of running outsourcing, hosting, and application management services. We offer industry-leading e-business solutions and integrated software environments, we support open standards across our product lines, and we have invested in utility computing, Web services, autonomic computing, and grid computing to ensure that we have the right tools, experience, and know-how.

IBM has the IT expertise, strong lineup of products and services that support an on demand operating environment, and deep industry experience to help customers transform their businesses into on demand businesses.

10. Can I sell the idea of on demand to all my customers?

Any business can become an on demand business—and every business needs to begin building its capabilities today. The advantages of being on demand apply to any business of any size in any industry. The on demand idea is a framework around which to engage customers in conversations about and in planning where they want their businesses to go, and in how IBM technology and services can help them get there. Depending on a customer's business and infrastructure needs and what stage of e-business adoption it's in, there are IBM offerings appropriate to help it develop its on demand capabilities. Companies that develop on demand capabilities first will gain competitive advantage in the marketplace.

11. Are there differences in on demand offerings for medium and large companies?

IBM has offerings to help businesses of all sizes develop and deploy the business processes and technology infrastructure required to be on demand.

12. How can you determine if customers are good candidates for on demand?

Different companies are in different stages of e-business, but any company can become an on demand business. Assessment offerings are being developed to help determine where customers are in their adoption of e-business. Most are not yet in the on demand phase. This will set up specific steps they should take to develop on demand capabilities in business processes and technology infrastructure. Until the tools are available, an account planning session with the IBM customer team should help begin assessing where they are, what offerings are appropriate, and what next steps in the discussion with customers should be.

13. Will on demand offerings be different by industry?

On demand is about transforming the way entire companies and industries do business, so that the range of offerings that can help businesses become on demand companies is broad and deep. IBM currently has many offerings that support on demand capabilities. Some solutions apply across all industries, while others are specialized by industry.

14. What is grid computing?

See "[Grid computing](#)" on page 269.

IBMer can learn more at the following sites:

- ▶ [Developing Grid computing applications](#) (Article)
- ▶ [IBM Grid Toolbox](#) (alphaWorks® download)
- ▶ [Installing and configuring the IBM Grid Toolbox](#) (Free tutorial)

15.What are autonomic computing systems?

See “[Autonomic computing](#)” on page 269.

IBMer can learn more at the following sites:

- ▶ [Autonomic computing overview and resources](#) (Collection of resources)
- ▶ [Autonomic computing](#) (Web site)
- ▶ [Tivoli®: Autonomic computing](#) (Web site)
- ▶ [Administration made easier: Scheduling and automation in DB2 Universal Database](#) (Article, in PDF)

16.What is utility computing?

See “[Utility computing](#)” on page 270.

17.How does e-business on demand differ from the idea of outsourcing?

In the outsourcing model—or “utility-like” computing—companies can acquire business processes, services, applications, and infrastructure over a network and pay for what they use as needed. This can be an important component of the on demand operating environment, since it is one way a company can move its computing costs from fixed to variable. But this is only one of many tools a company can use to help it become an on demand business. The on demand idea is about transforming the entire way a company does business, affecting business processes and relationships as well as technology infrastructure.

18.Will grid computing play a role in the on demand world?

Grid computing can be an important part of the capabilities of an on demand operating environment. Grids can provide the base for a resilient infrastructure, and they can coordinate and hide the complexity of distributed, heterogeneous computing systems. They can reduce costs by pooling technology resources so that any member of the grid can tap into the full computing capabilities of the entire grid as needed, and they can improve the utilization of technology resources by deploying computing resources from across the entire grid to the areas that need them most.

19.What are the criteria for an on demand offering?

An offerings that either:

- a. Helps customers meet one of the key attributes of an on demand business (*responsive, focused, variable, and resilient*), or
- b. Supports one of the capabilities of an on demand operating environment (*integrated, open, reliable, and resilient*—in other words, utility-like).

20. Does IBM have specific offerings that make up an on demand package?

Yes. Becoming an on demand company requires profound changes in the way it does business—much more than any one offering can deliver. But there are specific offerings that can help companies build their on demand capabilities. There are offerings designed for specific industries, for small and medium businesses, and for particular information technology needs. You may already be familiar with many of these offerings. We need to make sure that individual offerings are discussed as part of a larger plan with customers to help them transform their companies into on demand businesses.

An on demand business requires a particular type of infrastructure that IBM is uniquely positioned to offer with its depth, breadth, technology, and industry expertise. This *on demand operating environment* is made up of three components—integration, virtualization, and automation. IBM software offerings play a critical role in each of these areas.

Integration creates business flexibility by connecting people, processes, and information—anywhere, any time, from any device. IBM software offers collaboration, business integration, and information integration solutions for e-business on demand.

Virtualization is the pooling of IT resources for simplified access, improved working capital, and optimized asset utilization. These resources include information and applications (IBM software play) as well as servers, storage, and networks (IBM systems play).

Automation allows cost reduction and increased business responsiveness through policy-based orchestration. IBM software provides solutions in the areas of availability, security, optimization, and provisioning.

21. What do we have to sell today?

Many of our offerings already meet the needs of on demand businesses. Currently there are more than 30 offerings that make up the on demand portfolio, and there will be more to come in the future. IBMers can click [here](#)⁴ for a complete on demand offering list.

22. How does this fit into IBM's Industry Value Project?

The Industry Value Project provides us with the line of business relationships that allow us to understand industry issues and respond with the best and most relevant offerings for customers in those industries. Every IVP industry has a solutions portfolio of offerings, many of which can help customers develop their on demand capabilities.

⁴ <http://w3.ncs.ibm.com/sitelet.nsf/ondemand/JSHW-5MZUEN?OpenDocument&page=MP>

23.What is the role of Business Partners in on demand?

IBM is putting a tremendous amount of demand generation behind this initiative, and partners are sure to benefit from it as well. We see more opportunities for partners to expand their solutions, services, and software offerings and team with other partners to deliver complete on demand solutions to customers. Specifically, our Solution Developer partners will find more opportunity for their solutions and our consultant and integrator partners will see a demand for their expertise in delivering business insight. Specific on demand offering partners can sell now include:

- JD Edwards 5 w/Blue Stack
- IBM and PeopleSoft Accelerated CRM
- eQ from IBM and QAD
- Start Now Collaboration Solutions
- Start Now Infrastructure Management Solutions
- WebSphere Portal Express
- e-Server Integrated Platform for e-business (Linux)
- Business Continuity and Recovery
- Infrastructure Solutions with Server Management (Server Management Services)

1.2 Portals

1. What is a portal?

Portals provide the user with a single point of access to a wide variety of content, data, and services throughout an enterprise. The content displayed in portlets on the portal page can be personalized based on user preferences, site design, and marketing campaigns.

Definition: A *portal* is an integrated and personalized Web-based interface to information, applications, and collaborative services

A portal delivers integrated content and applications, plus a unified, collaborative workplace. Indeed, portals are the next-generation desktop, delivering e-business applications over the Web to all kinds of client devices.

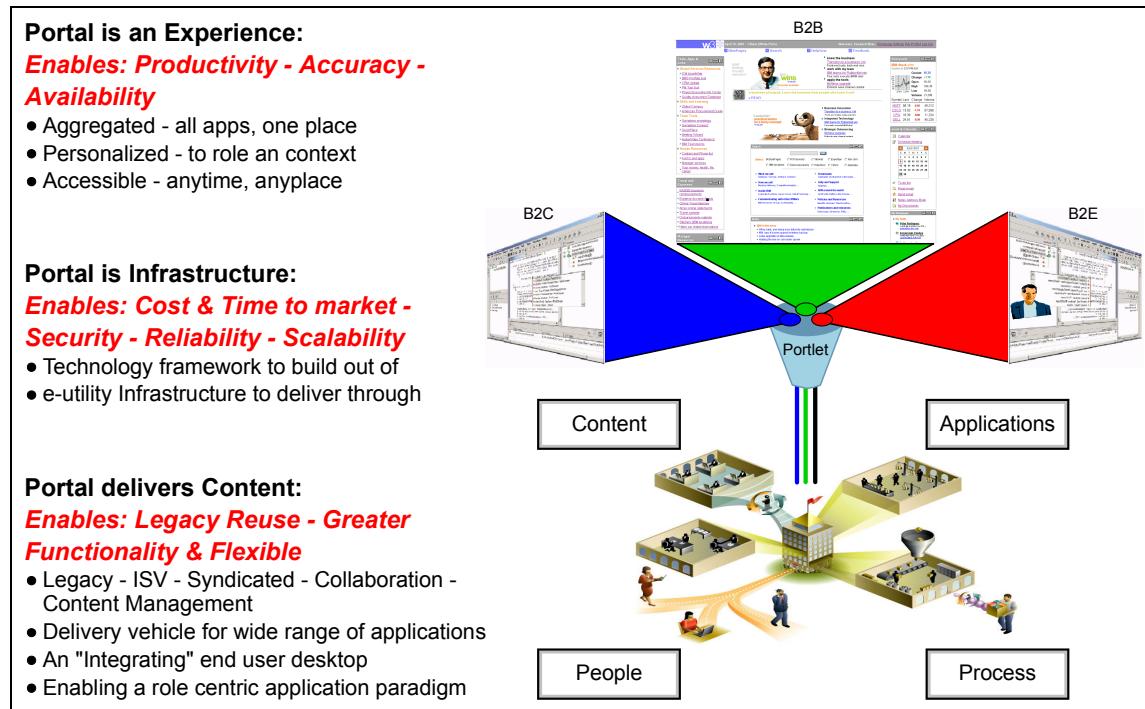


Figure 1-1 Portal aspects

Regardless of where the information resides or the format it's in, a portal aggregates all of the information in a way that is pleasing and relevant to the user. A complete portal solution should provide users with convenient access to everything they need to get their tasks done.

2. Do portals come in different flavors?

With many new products being announced, the marketplace has become very confusing. Indeed, any product or application that provides a Web interface to business content could be classified as a portal. For this reason portals come in many flavors and have many different uses (for example, B2E (Business to Employee), B2C (Business to Consumer) and B2B (Business to Business)).

Collaborative portals help business users organize, find, and share unstructured office and groupware content—for example, e-mail, discussion group material, office documents, forms, memos, meeting minutes, Web documents, and some support for live feeds. They differ from Internet and intranet portals not only in supporting a wider range of information, but also by providing a rich set content management and collaborative services. Content management services include text mining and clustering of related

unstructured information, information categorization to classify it and make it easy to find, summarization to generate abstracts for documents, publish and subscribe, finding people, and tracking expertise. Collaborative services allow users to chat, organize meetings, share calendaring information, define user communities, participate in net meetings, and share information in discussion groups and on whiteboards, etc. Collaborative portals are mainly used internally as a corporate facility, although B2B use is increasing.

Definition: *Collaborative portals* organize and track unstructured content and allow people to collaborate and share information

BI⁵portals provide executives, managers, and business analysts with easy access to business intelligence for making key business decisions. BI portals typically index business intelligence reports and analyses, canned queries, etc., associated with financial management, CRM,⁶ supply chain performance management, etc. They also provide seamless access to BI tools (reporting, OLAP, data mining), and packaged analytic applications and support alerting, publish and subscribe, etc. The main suppliers of BI portals are BI tool vendors who have extended their product lines to add a portal product. As such, these products are often limited to indexing intelligence produced by a single vendor BI tool set. These portals are mainly used internally as a corporate facility, although they are increasingly being deployed in trading exchanges—for example, to make demand intelligence available to suppliers to help optimize supply chains. Market demand for integration of unstructured information and business intelligence is causing BI and Collaborative portal vendors to provide support for both in a single portal offering, indicating that these two types of portals are beginning to merge.

Definition: *BI portals* organize and track intelligence for making key business decisions

⁵ Business Intelligence

⁶ Customer Relationship Management

Note: Types include:

- ▶ Procurement portal
- ▶ Self-service portal
- ▶ Business portal
- ▶ e-Commerce portal
- ▶ Sales support
- ▶ CRM, operations, and employee portals
- ▶ Consumer portal

These fall into the two main categories listed above.

3. Horizontal versus vertical portals

It is fair to say that most companies are looking at portals as a single user interface to everything—the new *Web top* and the access point to information and services from all computing devices. However, the way that portal technology has evolved and the difficulty surrounding the business case to access everything via one portal have meant that departments so far have started with smaller portal projects and grown into collaborative or BI vertical portals that are associated with their area. In addition, most IT organizations have restricted these to internal use.

Now there is a need to expand these vertical portals and to integrate them with e-commerce “standalone” Web sites, core operational applications, and other types of content to create the virtual enterprise. e-business portals are the new generation of horizontal portals that provide this solution because they offer a broad range of connectivity options and deep integration with enterprise systems. This includes e-business, operational, analytical, and collaborative applications plus data from disparate systems. They allow users to customize layout, personalize content and application components to suit their needs, and support access from pervasive devices. Furthermore, by providing an open portal framework that offers access to common portal services, e-business portals can provide support for deep integration of vertical portals on a common open framework.

Definitions: *Vertical portals* focus on specific application or departmental needs.

Horizontal portals connect to most sources of business content and provide common services to enable and integrate vertical portals

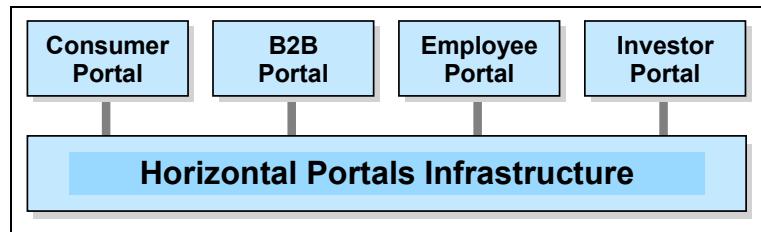


Figure 1-2 Horizontal portals infrastructure

IBM and some industry analysts have coalesced around the concept of horizontal and vertical portals. Horizontal portals are the primary infrastructure upon which a portal is built. Vertical portals are built upon the horizontal layer and represent a specific portal instance, usually defined by a major topic or domain.

WebSphere Portal's extensible framework allows the end user to interact with enterprise applications, people, content, and processes. They can personalize and organize their own view of the portal, manage their own profiles, and publish and share documents. WebSphere Portal provides additional services such as single sign-on, security, content management, search and taxonomy, support for mobile devices, and site analytics.

Portal Server is a *horizontal portal framework*, not a cute “flaming logo” vertical portal application. This is serious development tooling. Do not expect a quick and dirty project.

4. Why should my customer want to provide a portal?

a. Business drivers

Specific goals that the business is trying to achieve.

Three core business drivers:

i. Deep customer knowledge and mindshare

When a business wants to provide the “best customer service” experience and this is its primary driver for revenue, it must understand its customers and market well.

ii. Product leadership

These organizations want to achieve leadership from a quality and/or marketplace mindshare perspective.

iii. Transactional and process efficiency

Organizations that have identified increased efficiency in their internal processes will realize lower overhead costs and thereby increase revenue by attaining the highest possible efficiency in the transactions that take place between departments, divisions, employees, and external partners (for example, external suppliers who provide raw material for the products or services being offered).

b. Specific business goals

- Ease of use (single sign-on)
- Improve security
- Reduce TCO ⁷
- Shorten time-to-market
- Improve organizational efficiency
- Integration across multiple delivery channels
- Unified customer view across LOBs ⁸
- Support effective cross-selling
- Mass customization

A portal implementation requires the identification of the information desired, the audience for that information, and an analysis of its usefulness in order to fulfill the business drivers of an organization. Any given organization might need one, several, or all of these business drivers to meet its goals.

c. IT drivers

i. Reuse

Reusing existing IT assets such as programming code, existing applications, and existing data sources can reduce overall cost.

ii. Maintainability

Maintainability is a goal of the IT organization because shifting business goals often require adding to or diminishing functionality.

iii. Scalability

This enables the created solution to continue to work as the size of the load or problem being solved increases.

iv. Extensibility

The system design permits easier functional enhancement as the needs of the business change and/or increase.

⁷ Total Cost of Ownership

⁸ Lines of Business

d. Specific IT drivers

- Minimize application complexity
- Minimize TCO⁹
- Leverage existing skills
- Leverage legacy investment
- Back-end application integration
- Minimize enterprise complexity

Many of these IT drivers are focused on cost reduction through minimizing complexity.



Sidebar: Commerce portals

Many businesses strive to improve profitability, reduce cost, and strengthen customer relationships. By interacting more closely with customers and providing them with relevant content, these business goals can be achieved. For example, large-enterprise businesses may have several divisions or brands selling products and services on separate e-commerce Web sites. Often opportunities to increase revenues or reduce expenses are not leveraged. By creating a commerce-enabled portal, customers are able to access personalized content and applications in portlets defined by their choosing or by the business hosting the site.

Why commerce portals are used:

► To increase productivity and efficiency

Commerce portals provide the means for the user/ customer to have a single point of access to information directly related to their needs without having to hunt through Web site pages.

In business hosting, the commerce-enabled portal provides an infrastructure for managing customers, personalized content, and commerce. In addition, developers are provided common tooling for portal, commerce, and application development.

► To improve decision-making

A portal can provide the critical information needed by a specific user to make better decisions—for example, a buyer using a portal containing a portlet that provides alerts about products that need to be re-ordered once reaching a certain threshold.

► To strengthen competitive advantage and customer loyalty

Customers appreciate businesses that cater to their needs. By providing a single point of access using a portal, customer needs and desires for personalized content can provide your business with a competitive advantage. Content displayed in portlets can be aggregated and personalized based on user roles and personalization rules.

⁹ Total Cost of Ownership

► **Better reach and user experience**

Through the use of WebSphere Portal, multiple channels can access your Web site, including PC Web browsers, mobile phones, and wireless PDAs. The WebSphere Portal Server and WebSphere Commerce Server are capable of detecting what type of device is accessing the site and serving the appropriate markup language (HTML, XML, WAP WML) to display content in the portlets.

► **Better marketing opportunities and increased revenues**

A portal provides the environment to personalize content specific to a users interests. In the case of commerce-enabled portals, content displayed in a portlet can be personalized by user role or by personalization rules—for example, promoting the sale of jeans to all customers between 14 and 21. Personalization with portals provides a means for one-on-one marketing. Commerce-enabled portals also increase opportunities for cross-selling.

For most organizations in business today, commercial pressures demand faster decision-making, action taking, and increased productivity from employees amid a business climate drowning in a sea of content that continues to deepen and threatens to overwhelm them. This increasingly competitive climate has fueled an acute demand for a cost-effective, easy-to-use means of rapidly accessing relevant applications and information specific to job functions. Users also want the ability to share information and to communicate and collaborate easily with others. These basic needs apply whether a person works in a front-office, corporate, or back-office role. The same is true for business partners, prospects, and customers who are interacting with a company to supply products, obtain product information, purchase materials, or get customer service and support.

The challenge in providing such services is complicated by the fact that different people have different requirements. Each person needs access to different applications and different types of information distributed across internal and external systems on a range of disparate platforms. Access is also required from a variety of devices from various locations, with most people needing to collaborate with a unique community of people.

The solution is an e-business portal that provides users inside and outside the organization with a common integrated and personalized Web-based interface to the broad range of business content comprising information, business applications, and expertise. e-business portals are accessible from wired as well as wireless devices and are therefore network- and device-independent. In addition, they can support data and voice-based user interface. e-business portals also allow users to find people, collaborate with them, and uncover and organize business information for both *ad hoc* access and offline delivery. This section provides an introduction to portals, differentiates between types of portals, reviews requirements and architecture for an e-business portal, and takes a detailed look at IBM's portal strategy and product offering for both enterprises and service providers.

The personalized single point of access to all necessary resources reduces information overload, accelerates productivity, and increases Web site usage. But portals do much more. For example, they provide other valuable functions such as security, search, collaboration, and workflow.

A portal can help a business gain market share, retain existing customers, and reduce costs through the ability to target the delivery of information to specific user audiences.

Sales questions

- ▶ Do you want to reach new customers and new markets worldwide without building additional brick-and-mortar business?
- ▶ Do you need to reduce your cost of marketing, operations, and sales?
- ▶ Do you see a need to improve the productivity of your workers? Would you rather see them working on critical work rather than searching for and then sorting information? Would you like to see your employees working as a team, sharing knowledge on best business practices?
- ▶ Do you lack effective communication due to relevant information not at your fingertips?
- ▶ Do you want to optimize productivity and improve business processes?
- ▶ Do you currently need to access or route any paper-trailing documentation to complete your mission-critical work processes?
- ▶ Are your customers asking for the ability to view statements online and to view and pay bills electronically via the Web?
- ▶ Do you want to be paid faster for your delivered products?
- ▶ Do you want to optimize your inventory?
- ▶ Do you need new ways to improve customer service?
- ▶ Do you have a Call Center/Customer Service function?
- ▶ Would there be value in having the ability to find documents instantly to support questions from callers?
- ▶ Could you save time, improve customer satisfaction, and improve productivity with customer and product information available?
- ▶ Are employees finding different answers to the same question? Do workers in different departments working on the same business processes see the same information or different versions of that information?
- ▶ Is it easy (quick, secure, and providing an audit trail) to pass business processes from one completed task to the next?
- ▶ Are your suppliers asking for sharing business documents, orders, or specifications via the Web?
- ▶ Do you want to decrease your time-to-market?
- ▶ Would you like to create and evolve a robust online presence?

- ▶ Do your campaigns “speak” to your target customer? In other words, are they personalized?
- ▶ Do you have a cost-effective method of reaching your current customers in the manner in which they wish?
- ▶ Is the cost of customer acquisition exceeding your budget?
- ▶ Do you want to improve customer loyalty and retention?
- ▶ Can the sales reps access the product specifications, inventory, and price in real time while at the customer location?
- ▶ Does the sales person have a customer profile at his or her fingertips in order to identify up-sell or cross-sell opportunities?
- ▶ Are your sales personnel informed of new functions, pricing changes, and product releases in real time?
- ▶ Is there a way to remotely access information for training, asking questions, checking status, and processing orders?
- ▶ Do you need to cut cost-of-sales by having more self-service for the customer (for example, in ordering upgrades or new versions) while your personnel concentrate on the higher-profile sales?
- ▶ Do you have multiple employees working on the same business processes? Do they perform their tasks in an individual manner rather than using a standard method?
- ▶ Would you benefit from moving the business processes from your department to another department electronically and securely, with a paperless audit trail?
- ▶ Are you able to develop a department budget in an automated, interactive manner? Can you easily share budgetary items with other departments?
- ▶ Can users easily find the information they need?
- ▶ Does it take too much time to extract transaction files in order to populate the data warehouse for LOB queries? Do users complain that information is stale?
- ▶ Do you want to leverage the secure Internet for commerce transactions?
- ▶ Would you like the ability to reuse and re-purpose digital assets to save time and to generate business revenue?

5. Where are portals applicable?

There are three main types of e-business systems.

- a. B2E¹⁰ e-business covers most business activity, including CRM, financial management, HR, and supply chain management.
- b. B2B¹¹ e-business systems support supply chains, trading exchanges, and VARs.¹²
- c. B2C¹³ e-business systems support consumer e/m-commerce, customer self-service, and support. Each type of application has specific business needs.

For example, when attempting to grow customer mindshare and knowledge, a portal system can bring together the proper information tailored to the type of user that the business wants to target. Though this can be implemented in a number of ways, you should remember that when customers are made to feel that a business truly understands their wants and needs, the likelihood that they will be retained as customers greatly increases. Consequently, customer wants and needs can be met by product leadership, great customer service, or highly efficient transactional processes that support these. The components of personalization, multi-device type access, a presentation-rendering mechanism, and a business rules engine are combined with the ability to search and index content of various types and formats and to manage content through a workflow process that provides both content aggregation and a collaborative environment.

B2E e-business must allow all employees in front-office, corporate, and back-office functions to have personalized access to relevant, up-to-date business content, all through the same user interface. This includes access to operational, BI, and collaborative systems as well as external information. Mobile employees (for example, sales and field service employees) also require access to content from wireless devices and data synchronization services between mobile “lite” databases and enterprise systems.

B2C e-business requirements include consumer direct access to extensive content, front- and back-office operational applications, e/m-commerce applications permission marketing, search functions, and collaborative tools to support:

- Product information serving
- Product availability
- Secure purchasing
- Order status

¹⁰ Business to Employee

¹¹ Business to Business

¹² Value-Added Resellers

¹³ Business to Consumer

- Shipping status
- Self-service requests
- Marketing choice
- Reduced customer-service and support operating-costs
- Collaborative assisted service when needed (for example, via e-mail or voice)
- Access through browser and various other wired and wireless devices
- Confirmation of e/m-commerce transaction completion

B2B e-business (extended enterprise) requirements include personalized access by suppliers and resellers to operational, analytic, and collaborative applications and information as well as access to trading partner systems by employees. In addition, there is a need to define how business documents such as purchase orders, invoices, statements, and confirmations can be mapped between businesses. Application integration is required to integrate processes across businesses in procurement, billing, supply chain management, and distribution areas. Collaborative tools, self-satisfied service, and collaborative assisted service are also needed.

6. When are portals applicable?

Once an organization has determined that it needs to aggregate information, target that information to specific users, analyze the usage of information, and collect and manage information, it can use a portal to handle these requirements.

The IBM WebSphere Portal offering is for every large business that needs to reduce costs while improving revenue in this tight economy. Reducing employee costs while improving productivity and retaining satisfied employees is a challenge that most organizations face today. Building a sense of loyalty and greater degree of satisfaction with existing customers opens new opportunities for lower-cost up-sell or cross-sell revenue for the organization. Attracting new customers, with lower acquisition costs in marketing, can increase revenue as well as the company's reach. The WebSphere Portal offering can not only enable better customer care, acquisition, and retention, but it can also improve the business relationships with suppliers, business partners, channels, and service organizations.

7. What does a portal architecture look like?

Features include:

- a. **Presentation layer**—A Web user interface plus pervasive device support
- b. **Personalization**—The ability to serve dynamic response to the user based on personal profiles
- c. **Collaboration**—Tools that allow e-mail, team rooms, shared places, etc. to be exchanged

- d. **Portlets**—A framework for easily attaching software modules (portlets) and services
- e. **Applications and workflow**—Integration of legacy and new applications
- f. **Search and navigation**—Categorizing repositories of content and searching them for relevant information
- g. **Publish and subscribe**—The ability to author new content and publish it to subscribers
- h. **Administration and security**—Basic Web site services such as page design, performance monitoring, cluster services, and metadata management
- i. **Integration**—Metadata sharing, XML, connectors, standards, EAI

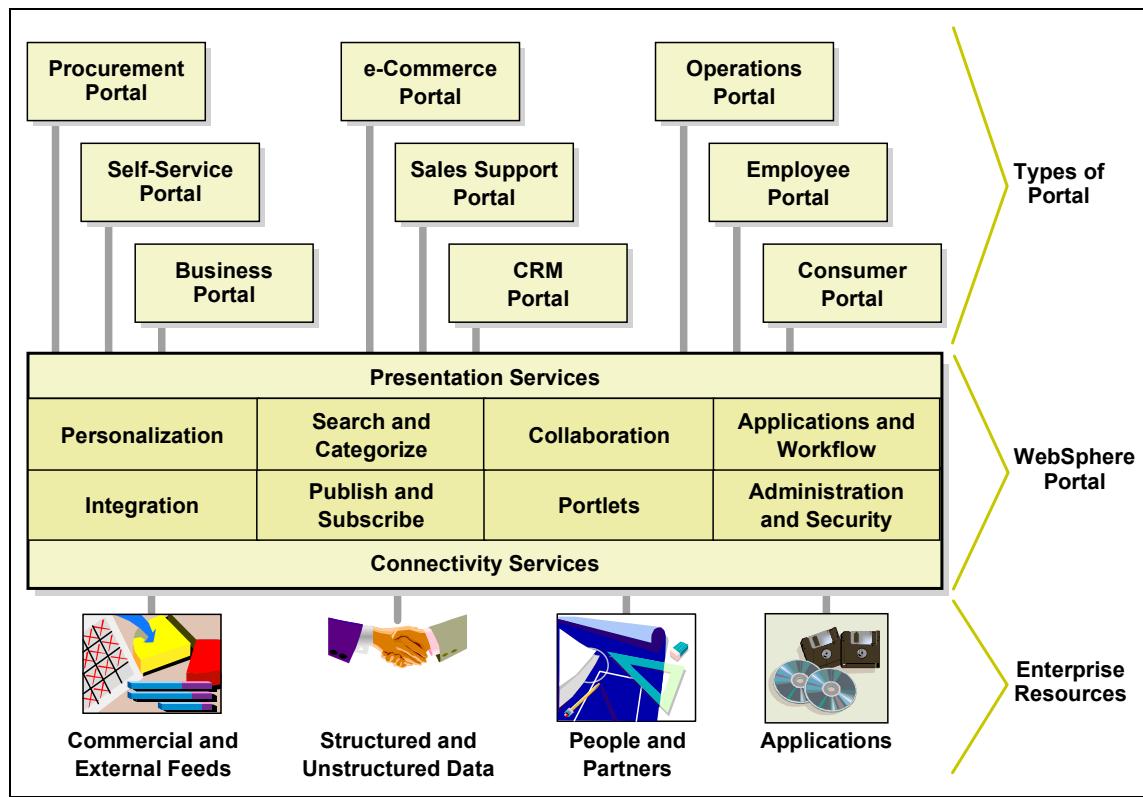


Figure 1-3 Portal services

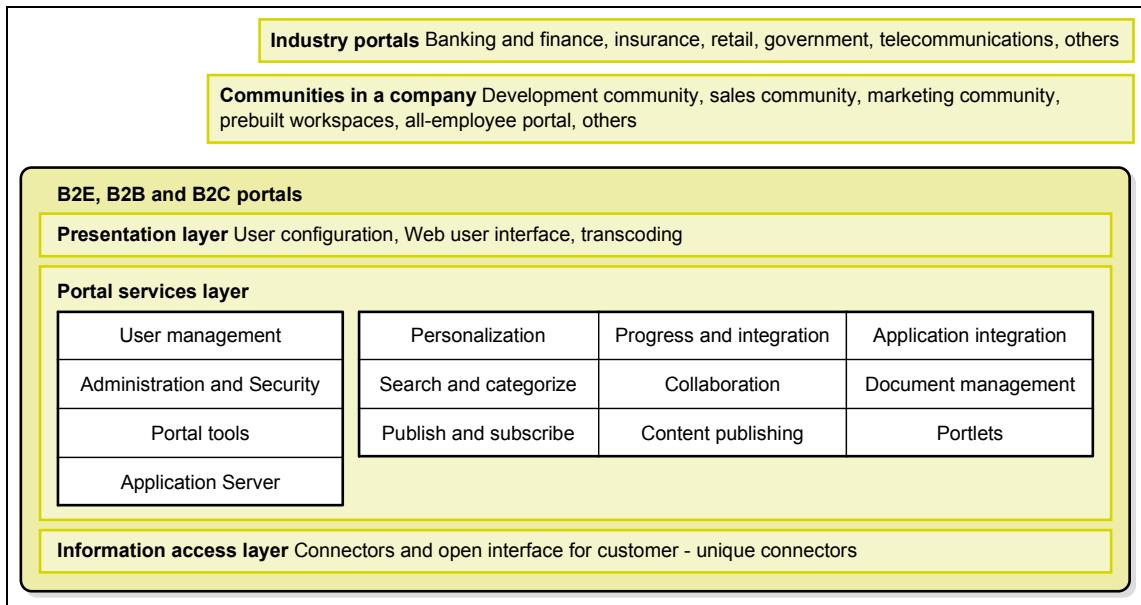


Figure 1-4 Portal layers

8. What is a portlet?

WebSphere Portal is a framework that lets you plug in new features or extensions called *portlets*. In the same way that a servlet is an application within a Web server, a portlet is an application within the WebSphere Portal. Developing portlets is the most important task in providing a portal that functions as the user's window to information and tasks.

9. Are there different types of portlets?

The following is from the portal overview presentation:

- COTS portlet**—This is a portlet provided by IBM or one of our business partners. No coding is required. Just add it to the portal server and configure it.
- Clipping portlet**—No coding required. Just configure the portlet through the Admin interface to point to a portion of any Web site.
- iFrame portlet**—Just configure the portlet and point to an entire Web site. All actions will be performed in the iFrame window. You can add SSO in many cases.
- XML/XSL portlets**—Can render any standard XML with and XSL template
- JSP portlets**—Can build a portlet entirely in JSP

- f. **iView portlets**—Can call a back-end SAP portal server iViews. Should be used for limited user sets (in other words, departmental) because of scalability and functionality subset. Use JCA for complex and robust tasks.
- g. **Custom portlet**—Can do anything in this portlet. Requires coding. Simple API that will be readily familiar to servlet developers.

10. Can I “portalize” an existing Web application?

Yes. To portalize an existing Web application, you need to transfer the business logic in your servlet to that in the portlet. For more information, IBMers should see “How to Portalize Your Existing Web Applications” at <http://w3.itso.ibm.com/itsoapps/Redbooks.nsf/RedbookAbstracts/tips0028.html?Open>

11. Leading the customer to portals?

Give the portal executive overview presentation, stressing the business value and positioning the portal vision of dynamic workplaces

Most companies start with a B2E implementation, then scale to business processes and data integration to get started on a B2E portal:

- a. Take an inventory of *assets, users, and projects*.
- b. Take the user's perspective.
- c. “Model office”
 - i. What would the user like to see on his or her desktop—data, applications, connection to people, and network?
 - ii. Select an initial “end-to-end” set of assets (for example, for a particular user task, have all the people, data, and applications selected for putting on the desktop)
 - iii. Prototype and refine
- d. Focus on business value to be delivered by the portal.

12. What is the value of IVP and portal solutions?

IBM's value proposition:

- IBM has the most complete portal offering on the market
 - IBM unifies the best-of-breed products and services that it has to offer.
 - IBM has delivered on the vision and promise of a single portal platform.
 - Like no other vendor's, IBM portal solutions have the embedded capabilities to offer the most advanced collaborative features.
- IBM WebSphere Portal provides the most flexibility and choice
 - IBM portal offerings grow as business needs grow.
 - They satisfy enterprise B2E, B2C, and B2B needs.

- They are built on open standards (J2EE and Web services).
- They provide out-of-the-box integration for whatever your enterprise needs.
- They are positioned for expansion to meet the needs of the SMB¹⁴ market space.
- Their new platform support includes iSeries™, zSeries®, and Linux for zSeries.
- WebSphere Portal is the portal the industry has already begun to rally around. More than 50 companies have joined the IBM PartnerWorld® Portlet Provider Program.

13. Where do IBMers go for help?

- WebSphere Portal—
http://www-3.ibm.com/software/info1/websphere/index.jsp?tab=products/portal&S_TACT=102BBW01&S_CMP=campaign
- WebSphere Portal Enable—
<http://www-3.ibm.com/software/genservers/portal/enable/>
- WebSphere Portal Extend—
<http://www-3.ibm.com/software/genservers/portal/extend/>
- WebSphere Portal Experience—
http://tuscarora.raleigh.ibm.com/wps2.1/InfoCenter/wpf/high_welcome.html
- WebSphere Portal Express—
<http://www-3.ibm.com/software/genservers/portalexpress/>
- Portal development site—http://tuscarora.raleigh.ibm.com/port_dev.html
- WebSphere Portal Catalog—
https://www-3.ibm.com/services/cwi/portal/_pagr/105/
- WebSphere Portal Zone (internal)—
<http://submit.boulder.ibm.com/wsdd/zones/portal/>

14. How do portals enable on demand?

Portals provide users access to information on demand.

You don't want to rip and replace. Instead, link together your disparate, distributed, heterogeneous systems. You can do this using portals. Portals can ease your integration burden and tie new products and technologies into your existing infrastructure easily and at lower cost.

In your on demand environment, you must be able to add or reduce capacity quickly as the business requires. In today's heterogeneous environment,

¹⁴ Small and Medium Business

open standards with common interfaces let you choose function from a variety of vendors and snap it into the infrastructure. The IT industry is evolving into a culture of open standards.

Note: For more information on this topic, see Appendix B, “Using WebSphere Portal” on page 383.

1.3 Patterns

1. What is a pattern?

A *pattern* is a structured and formal approach for describing a reusable solution to a reoccurring problem. Patterns should be concise and specific, but not inflexible. A successful pattern not only is reused in addressing the same reoccurring problem, but is flexible enough to be customized to meet a slightly different but relevant challenge. Patterns are typically a collection of documented best practices and lessons learned from tackling similar problems.

The concept of using well-defined patterns originated in the building architecture and construction industry. *A Pattern Language: Towns, Buildings, Construction* by Christopher Alexander *et al.*, published in 1977, has largely been credited with introducing the concept of patterns.

Buschman *et al.*, the authors of *Pattern-Oriented Software Architecture—A System of Patterns*, identified patterns for system architecture at a higher level than the original design patterns. Their patterns are related to the macro design of system components such as operating systems or network stacks. Information technology architects, encouraged by the success of design patterns, and facing challenges in systematic and repeatable description of systems, have also explored the idea of architectural patterns.

2. Why patterns?

Deployment of pattern *type standards* has contributed to rapid advancements in the computer hardware industry. This success and the need for reusability and rapid software development gave rise to object-oriented software, design patterns, and component-based development.

The idea of design patterns has gained acceptance by software designers and developers because it enables efficiency in both the communication and implementation of software design, based upon a common vocabulary and reference. Design patterns:

- a. Extend reach and value to field
- b. Reflect important configurations for IBM products
- c. Enable architects to demonstrate IBM thought leadership

- d. Provide SWG ITAs with tools to facilitate...
 - “Top-down” industry solution design with clients
 - Skills transfer and assimilation related to Industry solutions as well as IBM products
 - Comparison and positioning of solutions and products
- e. Document and communicate value of IBM intellectual capital

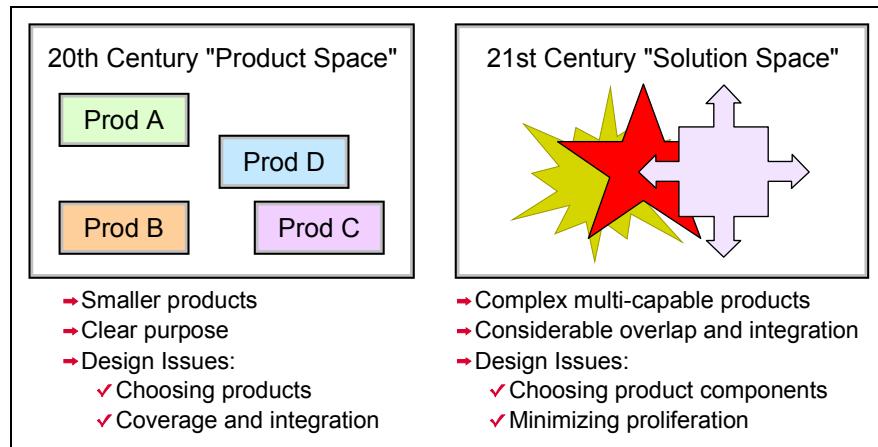


Figure 1-5 Product versus solution space

3. How can patterns help me?

Understanding the Patterns for e-business and specifically the Composite patterns is not always a straightforward process. In interpreting how to use the Patterns for e-business, it is best to start with how people in different roles might leverage these to explain and/or justify a particular solution.

Patterns leverage the experience of IBM architects to create solutions quickly, whether for a small local business or a large multinational enterprise. As shown in Figure 1-5, customer requirements are quickly translated through the different levels of Patterns assets to identify a final solution design and product mapping appropriate for the application being developed.

4. Who uses patterns and how?

In a portal implementation, the Portal composite pattern is the logical starting point. It identifies the Business and Integration patterns that make sense for the typical portal implementation. These roles are common in the IT industry, and each type of role will use and leverage patterns in a different manner.

- a. The **sales** role describes a person who is making the initial relationship with those organizations that might benefit from using and understanding patterns. This role can be filled by a person within an organization or someone from an external vendor¹⁵ who has the expertise to understand the business problems and issues that must be addressed. The

salesperson will use the patterns to begin the analysis discussion with the business-level stakeholders to understand the business drivers. They will start with the Business and Integration patterns and likely continue to the Application and Runtime patterns, showing how you can move from Business patterns to Application patterns to Runtime patterns.

During discussions with the business stakeholders, the initial team identifies high-level goals of the business and makes a determination that no specific Business or Integration pattern will address all the business drivers. At this point a Composite pattern makes sense. Especially when information-aggregation and other requirements are fulfilled by a portal, the Portal composite pattern is a good starting point. Refer to *Applying Pattern Approaches*¹⁶ for more details on how to use the Patterns for e-business in a sales role.

- b. A **project manager** will need an understanding of those patterns that have already been chosen so that a set of tasks can be derived. Priorities can be set because once the Business and Integration patterns have been chosen, the business and IT drivers are understood.
- c. The **architect** is the bridge between the business and technology domains. Once the business drivers are understood from the discussions with the sales role and business stakeholders, the architect can decide on likely IT drivers—namely, those goals upon which IT must focus in order to fulfill the business drivers. Combined discussions with both the business and IT stakeholders are important so that all can participate in the process of determining the final set of Business and Integration patterns, then decide on the Application patterns, and finally decide on the set of Runtime patterns. Once this is complete, the architect can derive an initial architecture¹⁷ for the implemented portal solution. Once design begins, it is the role of the architect to understand the “big picture” of the system and to make sure the proper components are given priority so that those business drivers that are most important are given top priority. A Composite pattern such as the Portal composite pattern saves the architect time by performing some initial “integration” work, bringing together various characteristics that are important to a typical portal implementation. Anything that speeds up the process (such as the Patterns for e-business assets) saves time and increases the chances for a successful implementation.

¹⁵ For example, IBM Global Services

¹⁶ SG24-6805-00

¹⁷ In other words, operational architecture and general architecture overview

- d. Although **developers** are generally tasked with very specific programming-level tasks, it's important for those in this role to understand the general thinking behind how the architecture was originally designed. This allows the team to leverage the focused technical knowledge of a developer (for example, an expert Java programmer) to understand how its tasks fit into the system and to alert it as to how its work might impact other components being designed. This role works to augment the architect role. The developer also uses the Application design and development guidelines provided by the Patterns for e-business to assist and speed up the application development cycle. These patterns are used to bring together the business and technical people in an organization. The intersection point of these two groups is the set of Runtime patterns that are detailed enough for developers and abstract enough for business people (because these Runtime patterns are far less complex than a portal or systems architecture diagram).

The Portal composite pattern is valuable because it performs some of the initial "integration thoughts" that lead to a typical portal implementation. Of course, the standard caution that "your mileage may vary" applies to the use of this Composite pattern, because each implementation will introduce some variation or implement a custom design. Using just one or two Business and/or Integration patterns might not address all the needs of both the business and IT drivers of the equation. The creation of the Portal composite pattern has brought together a combination of patterns that can jumpstart the design and analysis process. You can realize savings in time, people, and money by leveraging reusable assets such as the Patterns for e-business.

5. What is IBM's patterns for e-business?

The job of an IT architect is to evaluate business problems and design a solution based on the requirements and input of the customer.

The Patterns for e-business capture the experiences of IT architects by storing the knowledge gained in a repository of Patterns information publicly available on the Internet at <http://www.ibm.com/developerworks/patterns>.

The knowledge is further researched and tested to develop Business and lower-level patterns to save time and money on future customer engagements.

These approaches are further refined into useful, tangible guidelines. The patterns and their associated guidelines allow the architect to

- a. Start with a problem and a vision
- b. Find a conceptual pattern that fits this vision
- c. Define the necessary functional pieces that the application will need to succeed
- d. Build the application using coding techniques outlined in the guidelines

At the highest level, Business patterns define the possible business interactions required for a solution. The business function will typically fall into one or more of these defined Business patterns. A Business pattern describes the relationship between the users, the business organization or applications, and the data to be accessed.

Definition: Patterns for e-business are a group of reusable assets that can help speed the process of developing Web-based applications.

6. What assets are associated with IBM's patterns for e-business?

The reusable assets represented by the Patterns for e-business are broken down into the following elements:

- a. **Business patterns**—Used to identify the interaction between users, businesses, and data. Business patterns are used to create simple, end-to-end e-business applications.
- b. **Integration patterns**—Connect other Business patterns to create applications with advanced functionality. Integration patterns are used to combine Business patterns in advanced e-business applications.
- c. **Composite patterns**—Combinations of Business patterns and Integration patterns that have themselves become commonly used types of e-business applications. Composite patterns are advanced e-business applications.
- d. **Application patterns**—Define the high level application components and flow for refining the Business pattern.
- e. **Runtime patterns**—Used to define the logical middleware nodes used to implement the Application pattern.
- f. **Product mappings**—Define the actual products used to implement Runtime patterns on target platforms.

7. What are Business patterns?

Business patterns highlight the most commonly observed interactions between *users*, *businesses*, and *data*. They are the fundamental building blocks of most e-business solutions and describe the interaction between the participants in an e-business solution.

Definition: A *Business pattern* describes the relationship between the users, the business organizations or applications, and the data to be accessed.

The four Business patterns documented are:

- a. **Self-Service business pattern**—Also known as the *User-to-Business* or *U2B* pattern, this pattern captures the essence of direct interactions between interested parties and a business. These include customers, business partners, stakeholders, employees, and all other individuals with whom the business intends to interact. For simplicity, these interested parties are referred to as *users*. In this definition, business represents various types of organizations, including large enterprises, small and medium businesses, and government agencies.
- b. **Collaboration business pattern**—Also known as the *User-to-User* or *U2U* pattern, this pattern enables interaction and collaboration between users. It can be observed in solutions that support small or extended teams that need to work together in order to achieve a common goal.
- c. **Information Aggregation business pattern**—Also known as the *User-to-Data* or *U2D* pattern, this pattern can be observed in e-business solutions that allow users to access and manipulate data aggregated from multiple sources. This Business pattern captures the process of taking large volumes of data, text, images, video, etc., and using tools to extract useful information from them. These tools can personalize data to suit user preferences, distill summary information from large volumes of data, use algorithms to identify trends hidden in the data, or answer users' hypothetical "what-if" questions about potential business scenarios.
- d. **Extended Enterprise business pattern**—Also known as the *Business-to-Business* or *B2B* pattern, this pattern addresses the interactions and collaborations between business processes in separate enterprises. It can be observed in solutions that implement programmatic interfaces to connect inter-enterprise applications. In other words, it does not cover applications that are directly invoked through a user interface by business partners across organizational boundaries.

Table 1-1 The four documented Business patterns

Business Patterns	Description	Examples
Self-Service (User-to-Business)	Applications where users interact with a business via the Internet or intranet	Simple Web site applications
Information Aggregation (User-to-Data)	Applications where users can extract useful information from large volumes of data, text, images, etc.	Business intelligence, knowledge management, Web crawlers
Collaboration (User-to-User)	Applications where the Internet supports collaborative work between users	E-mail, community, chat, video conferencing, etc.
Extended Enterprise (Business-to-Business)	Applications that link two or more business processes across separate enterprises	EDI, supply chain management, etc.

It would be very convenient if all problems fit nicely into these four slots, but in reality things are often more complicated. The patterns assume that when broken down into their most basic components, most problems will fit more than one of these patterns. When a problem requires multiple Business patterns, the Patterns for e-business provide additional patterns in the form of Integration patterns.

8. What are integration patterns?

Complex e-business applications can be built by combining multiple Business patterns together. This is accomplished by using Integration patterns as the “glue” between Business patterns. Integration patterns differ from Business patterns in that they do not themselves automate specific business problems. Rather, they are used within Business patterns to support more advanced functions or to make Composite patterns usable by allowing the integration of two or more Business patterns.

Definition: *Integration patterns* allow us to tie together multiple Business patterns to solve a business problem.

The two documented Integration patterns that tie multiple Business patterns together are:

- a. **Access Integration pattern**—This pattern provides for front-end integration of multiple services and information through a common portal. It is responsible for handling multiple client device types, single sign-on, personalization, and providing a common look and feel for the application interfaces.
- b. **Application Integration pattern**—This pattern provides for the seamless back-end integration of multiple applications and data without the user having to access them directly.

Table 1-2 JThe two documented Integration patterns

Integration Patterns	Description	Examples
Access Integration	Integration of a number of services through a common entry point	Portals
Application Integration	Integration of multiple applications and data sources without the user directly invoking them	Message brokers, workflow managers

These Business and Integration patterns can be combined to implement installation-specific business solutions. We call this a *custom design*.

9. What are custom designs?

Business and Integration patterns can be combined to implement installation-specific business solutions. A custom design can also be a Composite pattern if it recurs many times across domains with similar business problems. For example, the iconic view of a Custom design in Figure 1-6 on page 36 can also describe a Sell-Side Hub composite pattern.

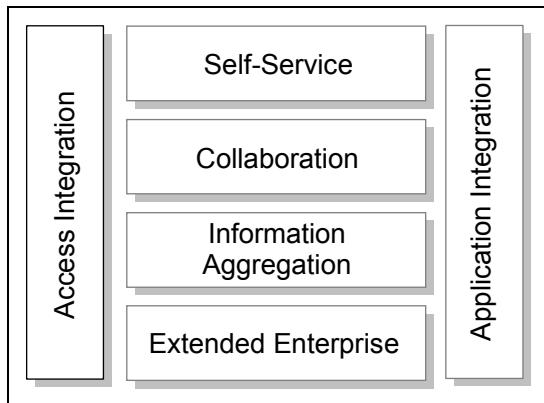


Figure 1-6 Iconic view of a custom design

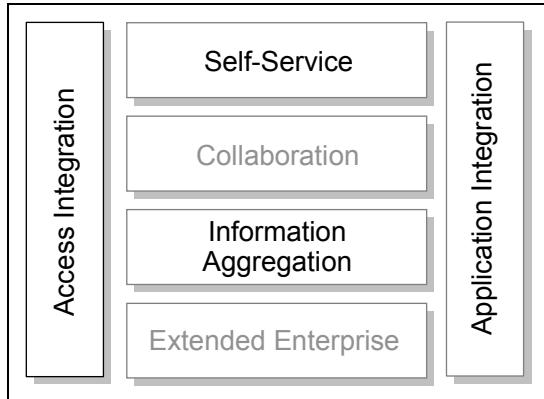


Figure 1-7 Custom design with Self-Service, Information Aggregation, Access Integration, and Application Integration

10. What are composite patterns?

Composite patterns combine Business and Integration patterns to create complex, advanced e-business applications. There are numerous potential combinations of Business and Integration patterns, but a solution design composed of these multiple building blocks is only considered a Composite pattern when it is recurrently employed to solve the problems of businesses across a wide range of industries.

Visually, the patterns shown in Figure 1-7 can represent either a custom design in a particular installation or a common Composite pattern.

Note: Recurring custom designs can become a composite pattern. For more information on Composite patterns, refer to *Patterns for e-business: A Strategy for Reuse* by Jonathan Adams, Srinivas Koushik, Guru Vasudeva, and George Galambos.

Table 1-3 Composite patterns

Composite Patterns	Description	Examples
Electronic Commerce	User-to-Online-Buying	<ul style="list-style-type: none"> • www.macys.com • www.amazon.com
Portal	Typically designed to aggregate multiple information sources and applications to provide uniform, seamless, and personalized access for its users.	<ul style="list-style-type: none"> • Enterprise Intranet portal providing self-service functions such as payroll, benefits, and travel expenses. • Collaboration providers who provide services such as e-mail or instant messaging.
Account Access	Provide customers with around-the-clock account access to their account information.	<ul style="list-style-type: none"> • Online brokerage trading apps. • Telephone company account manager functions. • Bank, credit card and insurance company online apps.
Trading Exchange	Allows buyers and sellers to trade goods and services on a public site.	<ul style="list-style-type: none"> • Buyer's side - interaction between buyer's procurement system and commerce functions of e-Marketplace. • Seller's side - interaction between the procurement functions of the e-Marketplace and its suppliers.
Sell-Side Hub (Supplier)	The seller owns the e-Marketplace and uses it as a vehicle to sell goods and services on the Web.	www.carmax.com (car purchase)
Buy-Side Hub (Purchaser)	The buyer of the goods owns the e-Marketplace and uses it as a vehicle to leverage the buying or procurement budget in soliciting the best deals for goods and services from prospective sellers across the Web.	www.wre.org (WorldWide Retail Exchange)

The makeup of these patterns is variable in that there will be basic patterns present for each type, but the Composite can easily be extended to meet additional criteria.

11.What are Application patterns?

Application patterns break the application down into the most basic conceptual components, identifying its goal.

12.What are Runtime patterns?

The Application pattern can be further refined with more explicit functions to be performed. Each function is associated with a runtime node. In reality these functions or nodes can exist on separate physical machines or may co-exist on the same machine.

13.What are product mappings?

The last step in defining the network structure for the application is to correlate real products with one or more runtime nodes.

14.What are the steps for using IBM's Patterns for e-business?

- a. **Selecting the Business pattern.** When faced with the challenge of designing a solution for a business problem, the first step is to take a high-level view of the goals you are trying to achieve. A proposed business scenario should be described and each element matched to an appropriate Business pattern. You may find that the total solution will require one or more Business patterns.
- b. **Selecting the Application pattern.** Application patterns break the application down into the most basic conceptual components, identifying the goal of the application. They do not show middleware, files, or the databases where Web pages might be stored.
- c. **Selecting the Runtime pattern.** The Application pattern must be underpinned by middleware functions. Each function is associated with a runtime node. In reality, these functions or nodes can exist on separate physical machines or co-exist on the same machine. In the Runtime pattern, this is irrelevant. The focus is on the logical nodes required and their placement in the overall network structure. For example, let's assume that our customer has determined that its business requirements fit one of the defined e-Marketplace composite patterns and that it's also selected the required Application patterns most descriptive of the scenario. The next step is to determine the appropriate Runtime pattern for the Application pattern.
- d. **Selecting the product mapping.** The last step in defining the network structure for the application is to instantiate real product names and versions with one or more Runtime pattern nodes. The product mappings are oriented toward a particular platform, though it is more likely that the customer will have a variety of platforms involved in the network. In this case, it is simply a matter of mixing and matching.



Portal solutions

The term *solution* can be used to describe both the answer to a problem and the method for solving it. This chapter will explore how both meanings apply to Web-based portal solutions.

2.1 Introduction

Portals have been around since the early days of the World Wide Web. Whether they knew it or not, probably everyone who has ever used the Internet has used a portal. Generally synonymous with *entrance* or *gate*, a portal is often the face we see when we use a browser to visit a company's Web site, whether to search for a book or to trade a stock online. Yet as common as portals are, their name is perhaps one of the most abused and misunderstood words in use today. This section attempts to clear up some of this confusion by defining a portal in terms of its general characteristics and uses.

Definition: A *portal* is an integrated and personalized Web-based interface that provides the user with a single point of access to a wide variety of data, knowledge, and services—at anytime and from anywhere using any Web-enabled client device.

2.1.1 What is a portal?

Simply stated, a portal is a single interface that provides convenient access to everything a user needs to get the task done, regardless of where it exists.

Whether to search for and buy a book, access an account balance and make a transfer, or update your personal information in the HR system at work, the portal brings everything together in one virtual place.

The fundamental characteristics of a portal:

1. Information aggregation
2. Targeted and personalized information
3. Accessibility
4. Single sign-on

Similar to a workstation desktop, a portal displays a variety of information and services in a single, consistent, user-friendly interface. Sometimes referred to as a *Web top*, a portal can be the major starting point or anchor site that users visit when they connect to the Web. Though unlike a traditional desktop, a Web-based portal is accessible via a wide range of Web-enabled client devices.

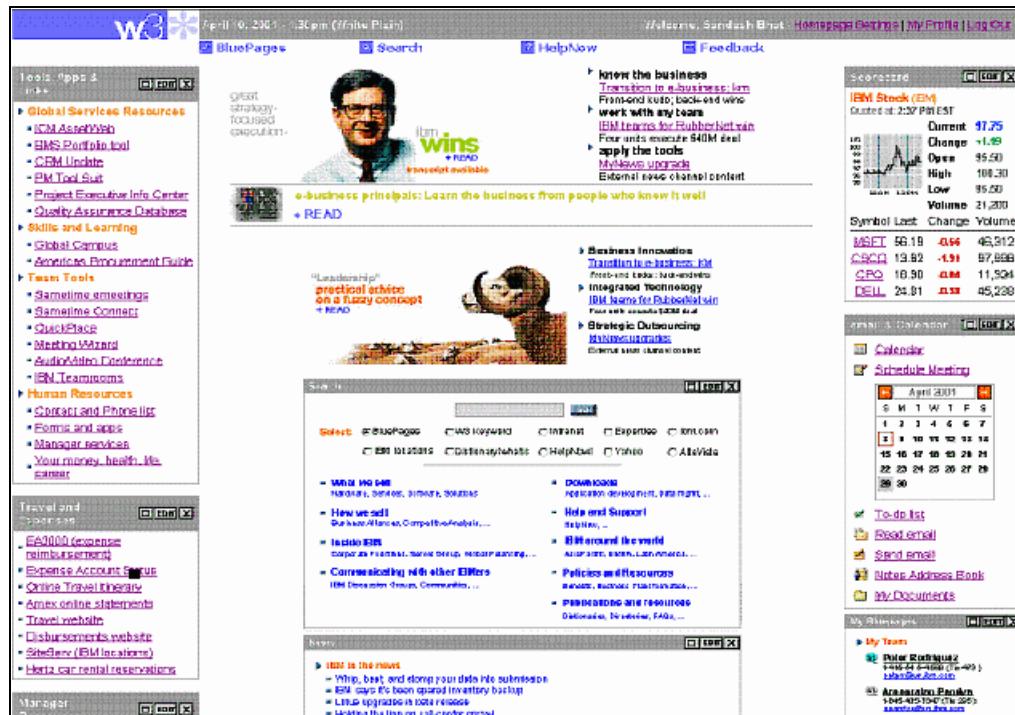


Figure 2-1 An example of a portal on a typical Web browser

What distinguishes a portal from a Web site?

1. A single point of access to all resources associated with the portal domain
2. Personalized user experience
3. Federated access to hundreds of data types and repositories, both aggregated and categorized
4. Collaboration and user integration capabilities
5. Integration with applications and workflow systems

2.1.2 Types of portals

An organization has many types of users that rely on its information and services. Customers, partners, and employees each have specific and often diverse needs. To address these needs, many different kinds of portals have been implemented. These can be categorized as follows:

Business-to-Consumer (B2C)

This type of extended enterprise portal (extranet) is associated with CRM¹⁸ and provides consumers with direct access to a variety of content—for example, product manuals and availability or price lists. Portal customers might also purchase products, check order status, and communicate with customer support. Like any other portal, a B2C portal is usually tailored to match customer needs.

Business-to-Business (B2B)

Another type of extended enterprise portal, B2B portals participate in supply chain management (SCM) by providing personalized access to business information by suppliers, resellers, and distributors. A typical B2B portal might provide a business partner with access to purchase orders, invoices, statements, and confirmations. Application integration is also required to integrate business processes in procurement, billing, manufacturing, and distribution areas.

Business-to-Employee (B2E)

B2E portals (also known as *intranet portals*) generally serve as a means to aggregate and disseminate corporate information and services to an organization's employees. There are two basic types of B2E portals:

- ▶ **Employee portals** provide access to relevant content such as company news, HR information, search engines, sources of expertise, reports, and other types of information generally applicable to all employees. These portals can enable employees to communicate and collaborate via chat rooms, discussion groups, etc. Typically, an employee portal also allows for

¹⁸ Customer Relationship Management

self-service, where an employee can sign up for classes or benefits, change personal information, etc.

- ▶ **Knowledge worker portals** are aimed at a particular role or set of roles such as sales. These portals often integrate content in order to support a particular process or processes. For example, an automotive technician might require resources from a number of applications such as service history, scheduling, or parts availability.

Public or mega

Sometimes called *Internet portals*, these portals are focused on addressing large audiences. There are two major types of public portals:

- ▶ **General public portals** address the entire Internet as opposed to a specific community (for example Yahoo, Google, and Excite).
- ▶ **Industrial or vertical portals** are focused on specific narrow audiences, (such as retailers, manufacturers, or finance).

It is important to recognize that a specific portal solution can be comprised of multiple types of portals blended into a hybrid solution. Equally important is that just as an organization has many types of users, it can have many different portals to support these users.

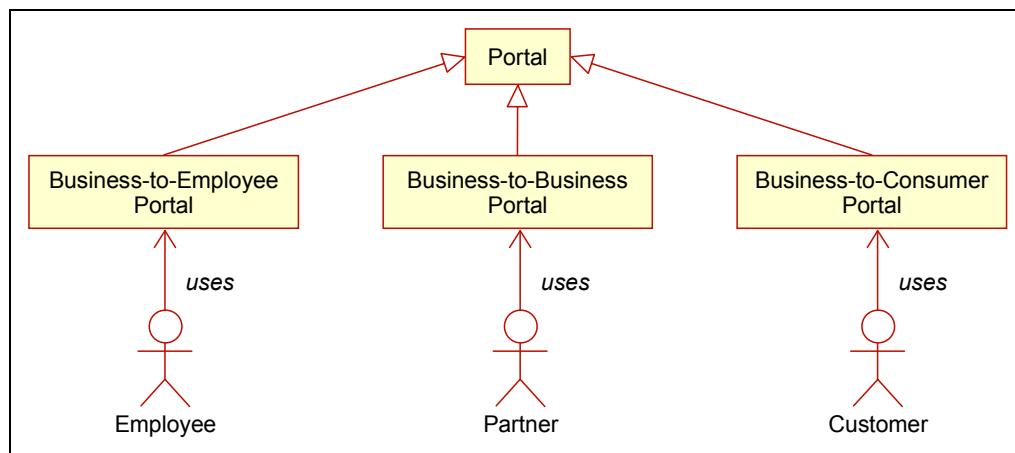


Figure 2-2 Categories of portals and who uses them

2.1.3 A brief history of portals

If we take a step back in time to the original PC days when each application took up the entire screen and used all of the computer's resources, the advent of Microsoft Windows revolutionized the way we interacted with our desktops. A

user no longer had to close one application to interact with another. Each application's content was aggregated to the desktop. This same kind evolution is now taking place on the Web with portal technology.

First-generation portals

The first portals, known as *first-generation portals*, were focused on providing static Web content, Web documents, and live feeds. Examples of first generation portals are Yahoo, Lycos, and Excite. In the corporate environment they had a similar objective—providing a single interface to corporate information distributed throughout the enterprise. These typically contained information such as company news, employee contact information, company policy documents, and other key Web links.

Second-generation portals

Second-generation portals focused on specific information and applications. They incorporated the notion of providing services along with the first-generation idea of providing content. Another key feature of second generation portals is *collaboration*. Collaboration portals provide the ability for teams to work in a virtual office. They provide content management services (the mining and organization of related information) along with collaborative services that allow users to chat, e-mail, share calendars, and define user communities. Collaborative portals are typically internal corporate portal installations.

Third-generation portals

Third-generation portals are intended to address full-function e-business. They take portals beyond the corporate boundaries for use by employees, suppliers, and customers. A significant addition to this generation is the integration of application servers. This means they provide a single point of integration for content and applications as well as collaborative services. They also provide access from multiple types of devices to address the diverse user communities in need of services. They offer the richest set of content and application choice through a single user interface to a diverse community, including browsers and pervasive devices. They also provide automated personalization via rules engines. The key to their further evolution is their open framework for common services.

2.1.4 Functional aspects of a portal

Although portals come in a wide variety of shapes and sizes, a few common features can be distilled:

- ▶ A portal aggregates information and services into one place
- ▶ It can be personalized for a group or individual
- ▶ It is accessible at anytime from anywhere

A portal aggregates content and integrates service

The stated purpose of a portal is to provide users with convenient access to everything they need to get the job done. This includes both information and services—regardless of where they originated.

Typical types of information provided by a portal

- ▶ **Structured data** is data that has been organized (often hierarchically by keywords) in such a way as to facilitate easy searching. A library card catalog is a good example. Structured data often includes reports, analyses, canned queries, and other types of “business intelligence.”
- ▶ **Unstructured data**, often hard to search, is unorganized data residing outside a database. Unstructured data can be text, audio, video, or graphics and take the form of office documents, memos, e-mails, meeting minutes, or any other such “knowledge.”
- ▶ **Syndicated content** is commercially available material (such as news reports, stock quotes, cartoons, and trivia) supplied specifically for the purpose of reuse and integration with other material.

Typical services provided by a portal

- ▶ **Collaborative services** (also known as communication services) allow people to chat, locate expertise, share calendars, participate in discussion groups, use whiteboards, etc.
- ▶ **Content management services** provide search, tracking, and data mining capabilities.
- ▶ **Self-service** (also known as transactional services) enable users to “interact” with systems directly without going through an intermediary such as a customer representative or salesperson. This allows users to buy products, schedule meetings, check account balances, enroll for classes, etc.

A portal can be personalized and customized

One of the more interesting aspects of a portal is its ability to be different things to different people. One of the ways it does this is through personalization and customization.

- ▶ **Personalizing** a portal involves choosing what should be displayed (the content). Often this is chosen automatically based on business rules such as the user's role in an organization. For example, when salespeople sign into the system, they are automatically presented with a list of new products.
- ▶ **Customizing** a portal involves choosing how the portal looks (theme and skin), what the navigation model will be, and where on the portal screen the content should be displayed (layout). A portal can even be “branded” to appear different to different types of users.

Personalization and customization allow a portal to target a specific community of users (such as customers, partners, or employees). Some portals may even be “individualized” to the preferences of a particular user.

A portal is accessible

Since a portal is Internet-based, it is accessible at anytime from anywhere using a standard Web browser. The advent of numerous Web-enabled devices such as cell phones and PDAs has enabled the portal to be extremely versatile and useful in a variety of settings.

2.1.5 Technical aspects of a portal

Portals provide a secure single point of access to diverse information and applications, personalized to the needs of their users. In some respects, enterprise information portals, B2B marketplaces, employee work spaces, and public Web portals have common requirements. All of these require scalable infrastructure, a flexible and powerful presentation framework, and a framework for building portal components easily. Each requires a high degree of personalization so that the most relevant information is delivered to the user, enabling a more productive interactive experience and encouraging user loyalty to the portal.

Depending on the nature and sensitivity of the information, some portals may require a greater degree of security, including specialized forms of authentication and access control. Depending on the size of the user base, some portals might require very high availability and scalability. Consumer portals generally allow users to enroll themselves and manage their own accounts. Conversely, enterprise portals often require integration with existing user databases or enrollment systems.

The portal framework simplifies the development and maintenance of portal sites.

- ▶ The page structure is defined only once
- ▶ Portlets are defined independently
- ▶ Portlets can be changed without impacting the overall page design
- ▶ Targeting multiple browsers and mobile devices is made easier

Portlets

A key building block in most portal frameworks is the *portlet*. Portlets are Java-based reusable user interface components that process requests and generate dynamic content. Executing in a runtime environment called a *portlet container*, portlets present their content in a window-like display on a portal page. Similar to a window on a desktop, the portlet window has a title bar that contains

controls that allow the user to expand (maximize) and shrink (minimize) the application.

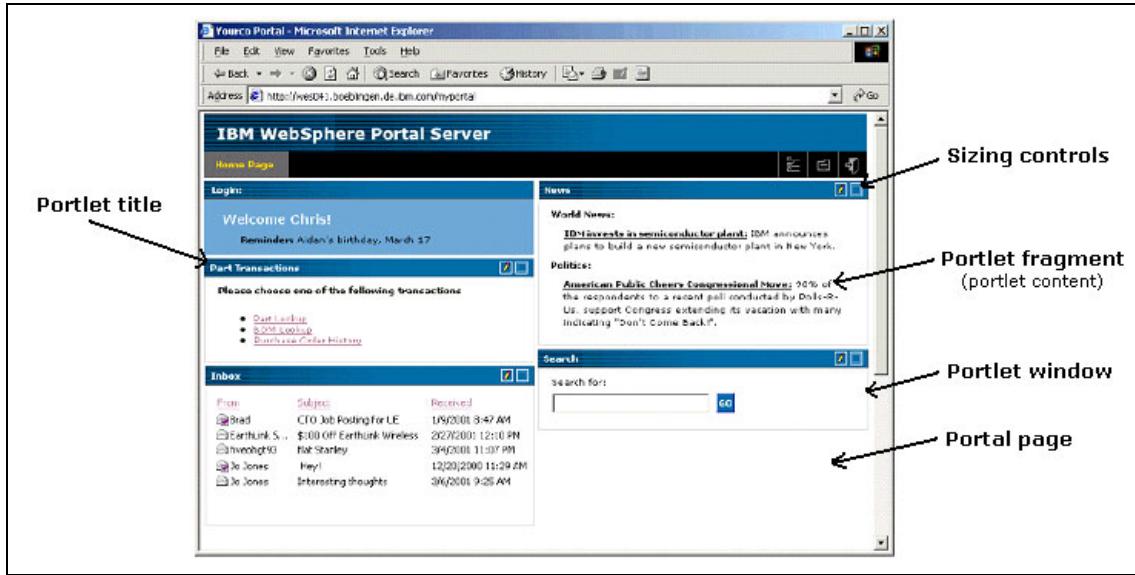


Figure 2-3 Elements of a portal page

Web clients interact with a portlet using the standard request/response paradigm. For a given request cycle, each portlet (identified via a configuration mechanism) generates specific content called a *fragment*. Each fragment represents a small portion of markup (for example, HTML or XHTML) that is aggregated with other fragments to form the complete response document.

Important: A portlet is visible on a portal page as a single small window. Each portal page can have many portlets. The portlet is the content inside the window, not the window itself.

Portlet container

Most portal frameworks provide the runtime execution environment for the portlets known as a *portlet container*. This is responsible for instantiating, invoking, and destroying the portlets it hosts in response to requests it receives from the portal server. Content aggregation is not a function associated with the portlet container, but rather with the portal or portal server.

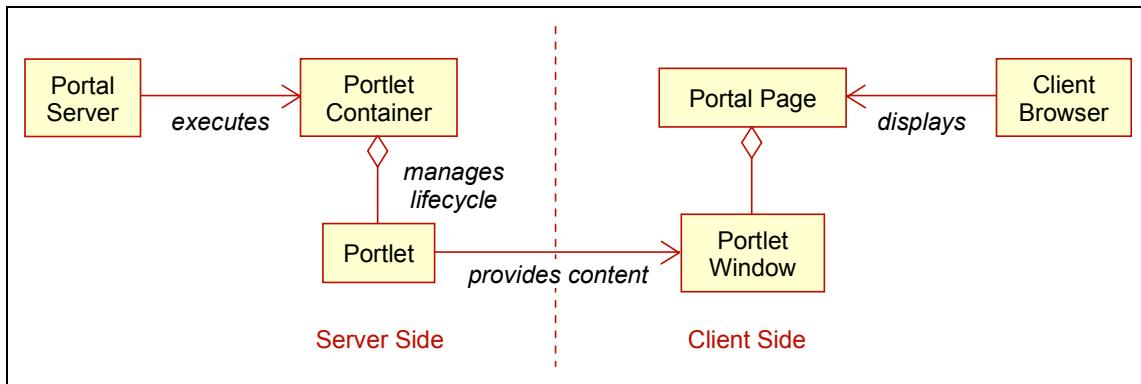


Figure 2-4 Depicts the context in which a portlet exists

Portal services

Portlets rely on the container to provide the necessary infrastructure to support a portal environment. The portal infrastructure provides the core sets of services required by the portlets.

- ▶ *Personalization services* enable the portlet to make use of rules engines and user profile information to modify content in order to make a user's visit to the portal more productive and satisfying.
- ▶ *Event notification services* enable portlets to respond to various requests in a fashion that is decoupled from the portal environment.
- ▶ *Communication services* provide portlet-to-portlet communication.
- ▶ *Content management support* facilitates connections to virtually any content or application source.
- ▶ *Search services* support heterogeneous searches across many data sources.
- ▶ *Collaboration services* enable users to communicate and participate in “communities of interest.”
- ▶ *User and group management services* allow users to enroll at the portal and to self-manage their own preferences and account information.
- ▶ *Page transformation services* provide support for a wide variety of client devices.
- ▶ Other services provide or manage:
 - User profile and other types of persistent data
 - *Security and access control services*, including user authentication and authorization
 - Performance monitoring, load balancing, and content caching

Portal server

The portal server is a specialized application server that provides business logic for a portal application. Typically built on top of an J2EE-compliant application server, the portal server provides development and runtime infrastructure for the portal. A portal server often works in conjunction with a Web server to process a client request.

Whereas a servlet can be viewed as a means of extending the functionality of the application server, a portlet can be seen as a way to extend the functionality of the portal server.

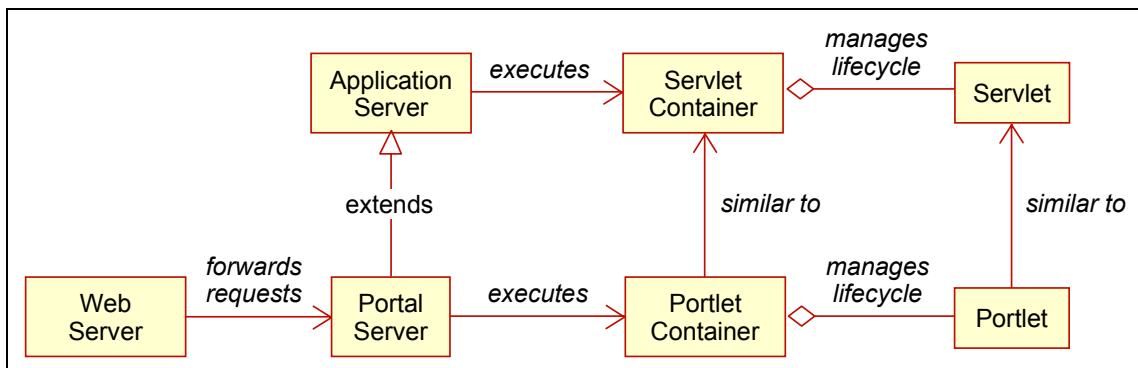


Figure 2-5 The portal server extends an application server to support portal applications

The following is a example of a typical portal request processing scenario. This scenario is initiated when the user requests the portal page from the client device.

1. A client device (for example, a browser or PDA) makes an HTTP request for the portal page to the Web server.
2. The Web server recognizes the request for a portal page and forwards the request to the portal server.
3. The portal server determines if the request contains an action targeted to a portlet on the portal page. If so, the portal requests the portlet container to invoke the portlet to process the action.
4. The portal server requests the portlet container to invoke the portlets associated with the portal page.
5. The portlet container requests each portlet associated with the portal page to render a fragment of dynamic content, and each does so.
6. The fragments are returned to the portal server, where they are aggregated to form the portal page.
7. The portal page is returned to the client device for display.

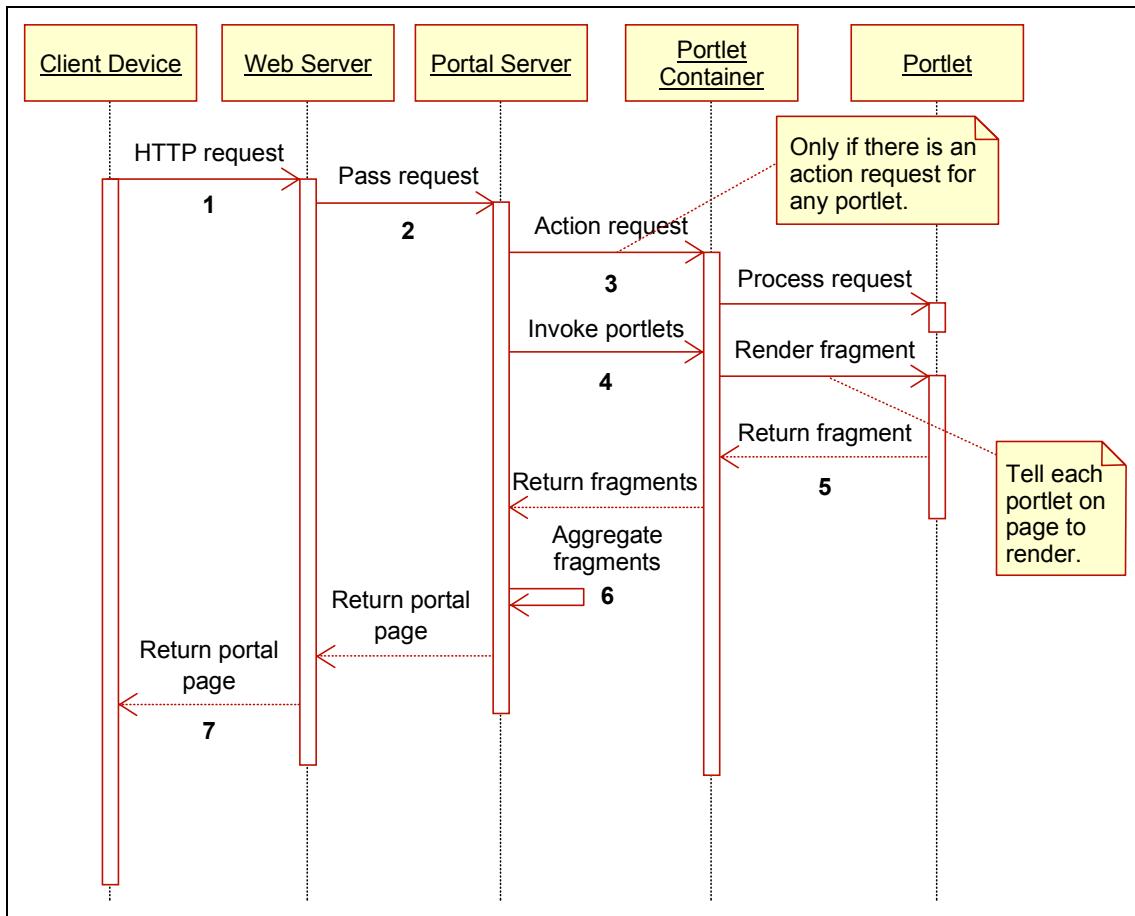


Figure 2-6 Typical portal page request processing scenario

2.2 Architecting portal solutions

An organization might want to consider using a portal if it needs to:

- ▶ Aggregate information and services
- ▶ Target information and services to a specific set of users
- ▶ Collect and manage information
- ▶ Analyze the use of information

This section will help you make an informed decision by showing you how to derive value from a portal solution and some important points to remember when

creating a portal solution. Should you decide to use a portal solution, it will provide you with a overview of the steps involved in developing a successful one.

Definition: A *portal solution*, though designed to result in many different portals, typically aggregates multiple information sources and applications in order to provide a single, seamless, personalized access to users via a wide variety of client devices.

2.2.1 Deriving value from portal solutions

A portal solution leverages various mechanisms (for example, content management, user interface formatting and display, and data aggregation) to bring together the appropriate information and existing systems to serve the goals of the business.

For example, when attempting to grow customer mindshare and knowledge, a portal system can bring together the proper information tailored to the type of user that the business would like to target. This can be implemented in a number of ways. The point to remember is that when customers are made to feel that a business truly understands their wants and needs, there is a high probability they will be retained as customers. Customer wants and needs can be satisfied through the business achieving product leadership, great customer service, or highly efficient transactional processes that support product leadership and/or customer service. The components of personalization, multi-type device access, a presentation-rendering mechanism, and a business rules engine are combined with the ability to search and index content of various types and formats as well as the management of content via a workflow process in order to provide both content aggregation and a collaborative environment. A portal can help a business gain market share, retain existing customers, and reduce costs through the ability to target the delivery of information to specific user audiences.

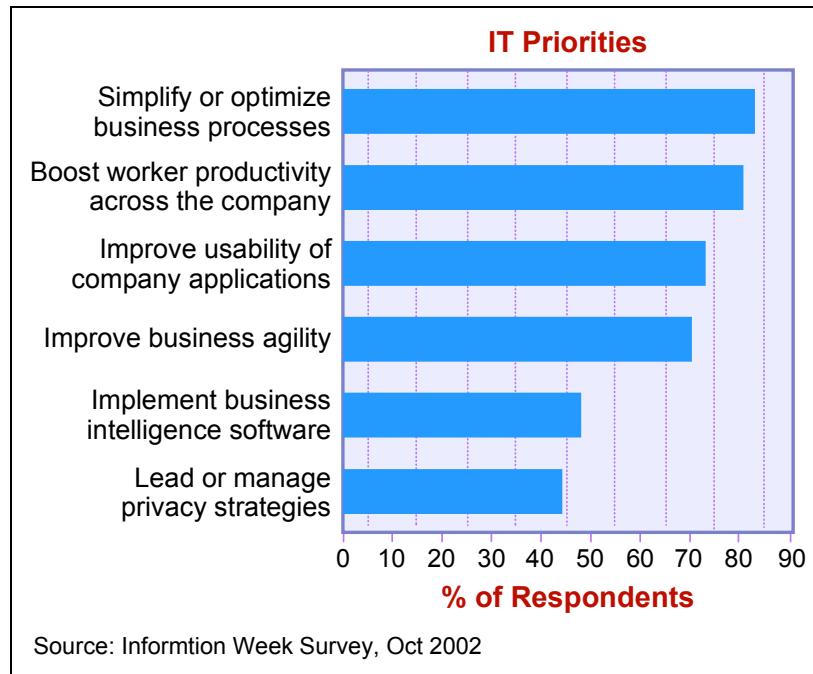


Figure 2-7 The most commonly reported IT priorities

Business drivers for choosing a portal solution

Business drivers are specific goals that the business is trying to achieve. In most cases, they have the ultimate goal of reducing costs, increasing revenue, or improving productivity. In fact, a business can be any type of organization¹⁹ that seeks to make the best use of its available resources and to determine if new resources are required. The design of a portal can help clarify these goals, and analysis of interactions with the portal can further define and enhance these drivers. There are various “paths” that can be followed to achieve the desired results, including:

- ▶ **Deep customer knowledge and mindshare**

This can be thought of as *customer intimacy*. When a business wants to provide the “best customer service” experience, and this is its primary driver for revenue, it needs to understand its customers and market as much as possible. It is therefore important to identify these types of customers when designing a portal so that once implemented, it can provide valuable knowledge about the habits of the targeted audience. This information can also be used to determine if the target audience is helping the business achieve its goals. Thus, an organization can increase customer retention

¹⁹ For example, manufacturing, research, or military

through deep knowledge of that customer, resulting in increased revenues through more efficient marketing practices.

► **Product leadership**

Some organizations want to be the best in their market for the products or services they provide. These organizations want to achieve leadership from a quality and/or marketplace mindshare perspective. One of the common methods for providing product mindshare leadership is by communicating to the targeted audience certain information about upcoming products or enhancements to existing products. If the business can identify other possible audiences to expand its customer base, this can contribute to product leadership as well. A portal can assist in disseminating both the technical and marketing information about the products or services provided, and this information can be tailored to specific user audiences.²⁰ In addition, the usage of the portal by these targeted audiences/customers can be analyzed to determine if marketing efforts are successful.

► **Transactional and process efficiency**

Organizations that have identified increased efficiency in their internal processes want to attain the highest possible efficiency in the transactions that take place between departments, divisions, employees, and external partners.²¹ A portal brings together information and access to that information into a single, aggregated view. This aggregated portal view of data provides just the information necessary for the person or entity to gain maximum efficiency in how tasks are accomplished. For example, in automobile manufacturing it is important for those on the assembly line to have focused technical information for the specific part of the vehicle they are assembling. They may also need information on parts that are related to their focus area because of the impact of given changes to their set tasks. As this information is being accessed, management can review how often and what specific parts of the information are being accessed, and thus determine if possible changes are needed to increase the efficiency of the assembly process.

Consequently, a portal implementation requires the identification of the information desired, the audience for that information, and an analysis of the usefulness of that information to arm the business drivers of the organization. Organizations may have only one of these business drivers, or there may be a combination of them that will help it meet its goals.

²⁰ As defined by demographic and “device type” information

²¹ For example, external suppliers who supply raw material for the products or services being offered

Concepts such as ease of use (for example, single sign-on, security, and reduced TCO²²) are all examples of specific, tactical goals of a portal implementation that will ultimately support the three core business drivers listed above. Following are additional examples of specific goals that can be used to achieve the ultimate drivers for an organization:

- ▶ Time-to-market
- ▶ Improved organizational efficiency
- ▶ Reduced latency of business events
- ▶ Adaptability during mergers and acquisitions
- ▶ Integration across multiple delivery channels
- ▶ Unified customer view across lines of business
- ▶ Support of effective cross-selling
- ▶ Support of mass customization (reducing the cost of customizing products and services)

IT drivers for choosing a portal solution

In all organizations, those concepts that drive IT to make decisions are ultimately driven by the needs of the organization at the business or enterprise level.²³

Each can be supported through the appropriate use of technologies that help implement the following goals:

- ▶ Minimize application complexity
- ▶ Minimize TCO
- ▶ Leverage existing skills
- ▶ Leverage legacy investment
- ▶ Integrate back-end application
- ▶ Minimize enterprise complexity
- ▶ Support maintainability
- ▶ Support scalability
- ▶ Support availability

Many of these IT drivers are focused on cost reduction through minimizing complexity. These can be further abstracted into five core IT drivers:

▶ Availability

The IT organization needs to have the solution available as defined in the business drivers. A portal implementation requires having the information customers want to see in a way they want to see it *when* they want to see it.

▶ Reuse

Reusing existing IT assets such as programming code, existing applications, and existing data sources can reduce overall cost. A portal implementation—

²² Total Cost of Ownership

²³ In other words, business drivers

specifically the Portal composite pattern—brings together various existing and new systems to construct an end-to-end solution.

► **Maintainability**

Maintainability is a goal of the IT organization because shifting business goals often require adding or deleting functionality. In addition, the sources of information available to a portal system may change. Thus, it is vital that a portal implementation be able to adapt to the changing environment by isolating different systems so that changes to one type of component will not affect other components that make up the portal system.

► **Scalability**

The *Portal composite pattern* is a “best mix” of nodes and components that lead to the Portal composite runtime pattern discussed in “Step 2D1: Seed the AOD (Logical Nodes) with logical system architecture from large reusable asset” on page 128. This Runtime pattern is a high-level representation of a portal architecture that separates the components so that each can be chosen for maximum scalability. Scalability is also important because the system should be designed and built only once and should be able to handle increased demands. This supports the general business driver of *reduced cost and operational efficiency*.

► **Extensibility**

Extensibility in a system design allows for easier functional enhancement as the needs of the business change and/or increase. Again, this IT driver supports the general business driver of reduced cost by enabling reuse of the same architected solution.

Benefits

Once an organization has determined that it needs to aggregate information, target that information to specific users, analyze its usage, and collect and manage it, a portal can be used to meet these requirements. Creating a portal architecture can produce the following benefits:

- A single aggregated view of content targeted to specific user types
- Ability to analyze usage patterns to make marketing efforts more efficient
- Ability to tailor the user interface to specific groups, enabling a focus on cultural, language, or nationality-based differences
- Single sign-on, allowing the user to save time and have access to information while decreasing the requirements for direct interaction with the organization in order to save money

Limitations

The creation of a portal is a complex undertaking. It requires the linking together of various, normally non-compatible systems to provide a single view of the information in an enterprise, impacting the organization as follows:

- ▶ Organizational changes
- ▶ Process changes
- ▶ Restructuring of existing data sources
- ▶ Rebuilding of some existing applications to support available connectivity options
- ▶ Detailed analysis of the various user groups that need to be supported (usually in much more detail than that which currently exists)

2.2.2 Portal solution best practices

Definition: A *best practice* is a technique, methodology, or innovative use of technology that through experience or research has reliably led to the desired result.

Though by definition somewhat subjective, true best practices tend to spread throughout the industry and become normal practices. In using best practices, a software architect is taking advantage of all available knowledge and technology in order to ensure success.

To create portal solutions that deliver true business value and help lead a business to becoming on demand, the software architect must embrace best practices whenever and wherever possible. This section identifies and describes several best practices for designing and implementing portal solutions.

IBM's Patterns for e-business

Best practices can come into play at virtually any phase in a solution's development. When it comes to architecting portal solutions, perhaps the most widely known best practice is IBM's Patterns for e-business.

When a company takes advantage of these documented assets, it can reduce the time and risk involved in completing a project.

Tip: For more information on Patterns for e-business, visit the Patterns for e-business Web site at <http://www.ibm.com/developerWorks/patterns/>

Start with proven solutions

Existing solutions in the form of *reference implementations* should be used whenever possible as the basis for architecting new portal solutions. By leveraging these implementations, software architects can provide greater value and reduced delivery times for their customers.

Tip: IBMers can learn more about reference implementations and industry solutions by visiting The BCS home page at <http://w3.ibm.com/services/bcs/> and following the links to Sectors and Industries and from there to the Industry Value Project or Integrated Industry Solutions.

Develop use cases

A use case methodology should be used to identify, clarify, and organize system requirements. In addition to helping the software architect gain a thorough understanding of the portal system's intended use, use cases can help the architect gauge and manage the overall scope of the system.

Tips: When designing a portal, it's important to remember...

1. Who are your users
2. What tasks they need to perform
3. The context in which they must perform those tasks

IBM's Rational® Software Site (<http://www.rational.com/>) provides a wealth of information, from developing use cases and documenting them with UML to using a full-blown unified development process such as Rational Unified Process® (RUP®).

Get buy-in

Developing a portal can be an extremely large undertaking. Simply managing the content in most modern portals takes a huge effort. So, before a software architect begins developing a portal solution, he or she must ensure that an executive sponsor and owner has been clearly identified.

The user community represents another group that must "buy in" to the solution before success can be assured. A "build it and they will come" approach is a ticket to disaster. Before a portal solution is deployed, it should be marketed. Providing education materials, introduction messages, preparatory e-mails, letters from executives, contact points, and frequently asked questions can dramatically aid user acceptance and help users overcome the fear of the unknown that often accompany new solutions.

Plan for maintenance

Today's portal implementations can be extremely complex IT environments, encompassing a wide variety of disparate applications, information, and hardware. Portals also have a tendency to grow over time, potentially doubling or even tripling in function and size. Successful solutions might grow even larger. A number of best practices for maintaining complex systems exist, and the software architect must take full advantage of these in order to manage this type of complexity and growth.

Customization and personalization

Though an inherent part of virtually all portal frameworks, use customization and personalization should not be overlooked. User acceptance of a portal can be greatly enhanced by even a small amount of customization (such as providing international language support). Enabling users to personalize a site by choosing content and layout can lead to a high degree of user satisfaction.

2.2.3 Steps for building a successful portal solution

The steps for building a successful portal solution can be simply stated as:

Plan

The *plan* phase formulates and documents an overall solution plan that should clearly link to the customer's business strategy and initiatives. For example, using a portal solution plays into companies' overall e-business strategies. A plan should detail the scope of the project, the customer's envisioned goals, and the business context (in other words, internal and external relationships).

Execute

The *execute* phase creates an architecture that translates business requirements into a portal application. As with any software application, the solution architecture should address both functional and operational domains. The *functional domain* addresses the business functionality of the solution. It describes the structure and modularity of the software components that will be used to realize business functions, the interactions between components, and the interfaces among them. The *operational domain* describes the system organization (such as hardware platforms, connections, locations, and topology), the placement of software and data components, the non-functional requirements (such as performance, availability, and security), and system management (capacity planning, software distribution, and backup and recovery). The architecture also needs to align the solution with the overall business strategy and provide a roadmap for its implementation.

Implement

The *implement* phase implements the solution developed and refined in earlier phases. This includes designing, developing, testing, and deploying the actual portal application. Typically, the software architect is not involved in this phase, and when so usually in a consultant role to ensure that customer expectations are met.

2.3 Portals and e-business on demand

As we have seen, portals have been around for years and are likely to remain for the foreseeable future. It appears that e-business is heading towards what IBM terms “the era of e-business on demand.” This section describes where portals fit into *on demand* and explains how they can help lead an e-business in that direction.

2.3.1 What is an e-business on demand?

IBM defines this as *one whose leaders can see and manage their company as an integrated whole*.

Throughout the business, previously isolated departmental operations must be transformed into business processes integrated across the company and in the way they are viewed by its customers.

An on demand business has four essential characteristics:

- ▶ **Responsive**—Intuitively responsive to dynamic, unpredictable changes in demand, supply, pricing, labor, and competition
- ▶ **Variable**—Flexible in adapting to variable cost structures and processes associated with productivity, capital, and finance
- ▶ **Focused**—Concentrated on core competency, differentiated tasks, and assets, with tightly integrated strategic partners
- ▶ **Resilient**—Capable of managing changes and threats with consistent availability and security

To attain these characteristics, companies need an operational environment that ties their business processes together.

2.3.2 Building the on demand operating environment

Integration and connectivity to this extent are accomplished through what IBM calls the *on demand operating environment*. This has four characteristics:

- ▶ **Integrated**—Far more than simply connecting disparate computing resources, the on demand environment must enable the integration of core processes and systems so that business can flow inside companies and across multiple enterprises.
- ▶ **Open**—As the basis for integration, open technical interfaces, and agreed-upon standards are essential.
- ▶ **Virtualized**—Through an interconnected consolidation of computing infrastructure, grid computing will allow distributed computing resources to be shared and managed as a single, large, virtual computer.
- ▶ **Autonomic**—Similar to the way the human autonomic system manages our vital functions, the enormous complexity of tomorrow's computing systems must rely on technology that manages itself.

The benefits of an on demand operating environment are enormous. Because business applications can be integrated according to open standards, it facilitates system interoperability; and as businesses prepare their internal systems to support processes across the company, they can take advantage of a growing virtualized grid of computing resources that allows them to reach more customers farther and faster. Finally, the complexity of these systems will be managed autonomically, thus relieving much of the human-intensive activity traditionally associated with large-scale business computing.

2.3.3 The evolution of e-business on demand

An e-business on demand evolves in phases. In each phase, the Internet transforms the business processes.

- ▶ **Access to digital information**—This phase is all about publishing content, mostly of the static “look-up” variety. Simple database queries allow us to check a bank account, check airline flight information, or see where our overnight package is. It's pretty easy to get in the game here. All an enterprise needs is a home page, and all the user needs is a browser.
- ▶ **Real transactions, real e-business**—Don't just look at your bank account—move some money. Don't just check a flight departure time—book your seat. Trade a stock, buy a book, apply for a loan, renew your driver's license, take a college course. Doing this requires more than a Web site—it requires behind-the-scenes integration of technologies and business processes.

- ▶ **The advanced stage of e-business**—In a fluid system of customers, suppliers, partners, and employees, the Internet is the primary way to communicate, transact, and connect. Business processes shift from manual to automated. A relationship could last only as long as a single transaction. The environment is real time computing. Here we form networked communities so that organizations can:
 - Create new products and services faster
 - Reach new customers and add new relationships economically
 - Dynamically change existing relationships
 - Simultaneously engage in multiple e-business models
 - Improve access to information by constituents involved in these relationships

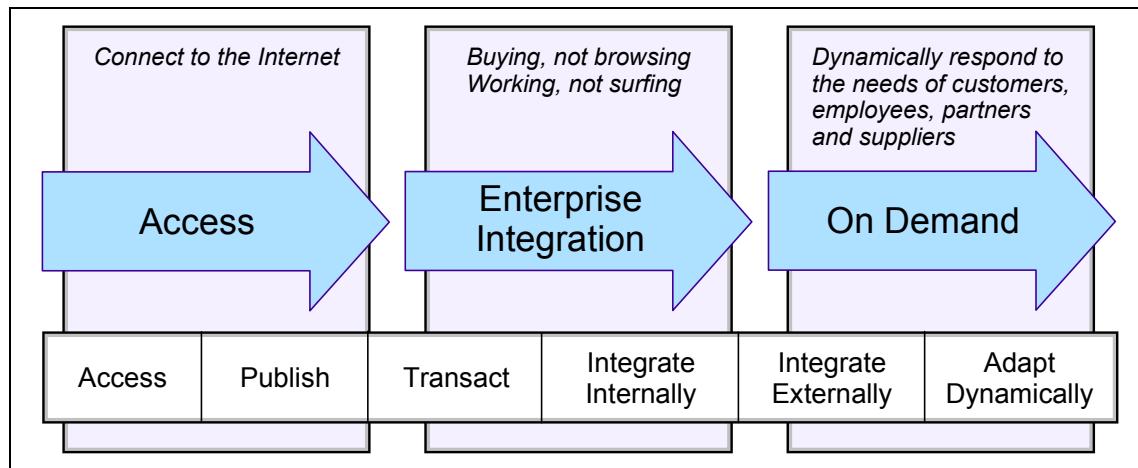


Figure 2-8 Steps required to become an e-business on demand

What's the return on investment for an enterprise that progresses through these phases? An increased share of customer spending, a better return on assets, new revenue opportunities, and better shareholder return.

Every industry segment has the common requirement to integrate end-to-end so that products, government and private sector services, invoices, images, decisions, and answers are all available on demand. Who will have the competitive advantage? The enterprises that get there first, with the software and service providers that know how to make it happen.

Any globally connected enterprise must have the ability to handle whatever comes its way, including changes in customer preferences or competitive actions, fluctuations in capital markets, labor situations, natural disasters, or political unrest. An advanced e-business must be able to respond to the

unpredictable and the unforeseen and never question the ability of the infrastructure to deliver.

In order to achieve this vision of “on demand,” cross-industry integration is necessary. Industry consortiums, open source communities, and standards bodies are busy forging the common language, constructs, and protocols that will be used. Companies have spent billions on existing IT infrastructure, and they need to leverage its power. They must be able both to extend and customize existing applications and to build and deploy new applications more quickly than ever before.

Which phase is your business in?

- ▶ About 75% of all businesses are in the *Access* phase. These businesses provide a basic Web delivery channel to their customers.
- ▶ Another 20% are in the *Enterprise Integration* phase. These businesses have identified key processes and are applying Web technologies to transform them.

2.3.4 How portals can lead to an e-business on demand

In the on demand world, business needs change more rapidly than in traditional models. As business conditions change, new opportunities will emerge and new threats will appear. In order to respond effectively, a company must be able to:

- ▶ Assimilate information quickly and make rapid decisions
- ▶ Reform existing applications and create new ones in the blink of an eye
- ▶ Target specific users with focused information, products, and services

This is the type of world where portals excel!

Portals enable a business to be responsive and variable

Portals by their very definition are accessible at *any time* from *anywhere* and *on demand*, with the ability to aggregate and display all relevant information. This puts the portal in a powerful position to enable people to make decisions quickly and effectively.

Businesses must be able to modify both their internal and external applications in the shortest possible time. Portals are specifically designed to be “snapped together” easily. This allows existing portals to be quickly reformed. With hundreds of off-the-shelf portlets now available, new portals can be quickly created and deployed, *on demand*.

Portals provide focus

Portals can be tailored to target a particular group of users, with very specific information and services allowing the business to focus its attention where it makes the most sense.

Portals are resilient

In the face of availability and security concerns, portals can help a business become resilient. Since portals are fundamentally constructed from components,²⁴ this provides the portal with a high degree of scalability and deployment flexibility. Portals also give the user a single point of access that can be made highly secure.

Portals enable integration

Portals (more specifically, the B2E variety) are considered a user-role integration vehicle because they facilitate a role-centric application paradigm. In this model employees are divided up into subgroups based on the roles they play. By identifying the primary functions of a particular role, interactions with a particular business process can be easily addressed. In this way the portal eases the burden of business process integration by enabling a role-centric approach. The B2E portal is then used to allow all employees to have personalized access (based on role) to relevant, up-to-date business content through the same user interface. An integrated interface and resulting user experience can actually ease back-end integration pains by “buying time” until strategic integration plans can be executed.

Portals are open

Enterprise-level portal infrastructures such as IBM's WebSphere Portal Server are built using standards-based technologies such as J2EE and Java. Standards such as SOAP²⁵ and the soon-to-be JSR 168²⁶ focus on interoperability between portals and data sources.

2.3.5 Portals in an on demand operating environment

The most common types of business applications used by today's companies drive a variety of management capabilities—including CRM, ERM,²⁷ PLM,²⁸ and VCM.²⁹ These remain valuable resources. However, in the on demand era, these applications are not only integrated end-to-end across the company—they are

²⁴ For example, portlets, Web services, and applications

²⁵ Simple Object Access Protocol

²⁶ JSR stands for *Java Specification Request*

²⁷ Enterprise Resource Management

²⁸ Product Lifecycle Management

²⁹ Value Chain Management

also integrated with key suppliers and/or distributors outside the company, ultimately delivering greater business value to customers.

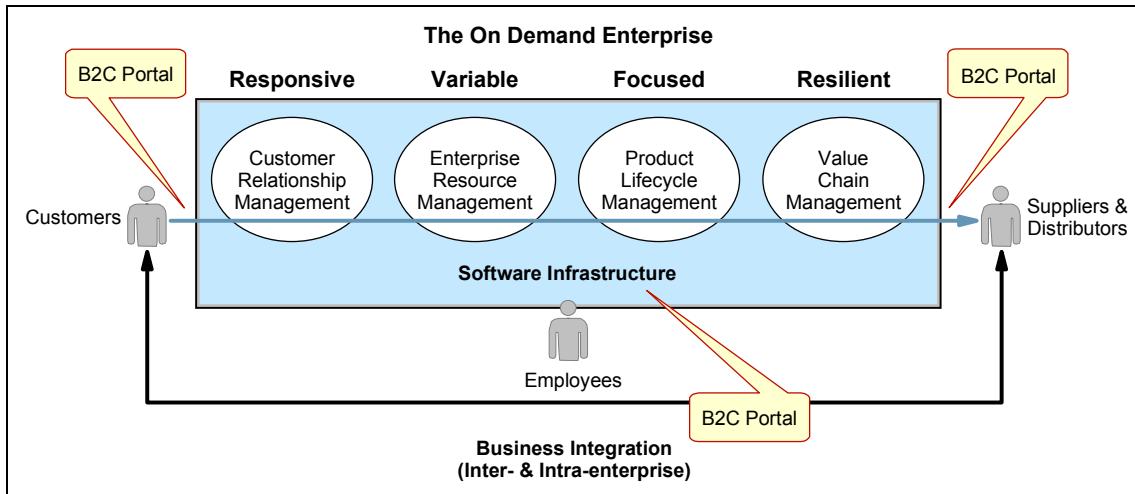


Figure 2-9 *The on demand enterprise*

2.4 Summary

Portals are the next-generation desktop, providing site users with a convenient, single point of access to a wide variety of content and services using all kinds of client devices over the Web. A portal aggregates information in a way that is pleasing and relevant to users, providing everything they need to get their tasks done.

A horizontal portal provides a common framework upon which to build, deploy, and integrate Web applications in a secure, reliable, and scalable manner. A portal delivers business value by leveraging existing Web infrastructure, allowing greater legacy reuse and increasing employee productivity. It also helps a business integrate processes, enabling role-centric application paradigms.



A recommended approach to architecting portal solutions

This chapter is a detailed guide on how to architect and design portal solutions for customers in a wide range of industries by leveraging large reusable assets, best practices, and experiences . In order to make this apply to real projects, we have embedded these practices and experiences into an industrial-strength methodology, including a step-by-step process for asset deployment. This guide weaves the process into the appropriate phases, tasks, and activities required to develop all required work products.

Our methodology focuses primarily on where and how to leverage these assets. As we lead you through our solution development lifecycle, we spend relatively little time on *implementation*, since the assets utilized there tend to be smaller and more focused,³⁰ and this phase generally much better understood and documented than the others. Also, please note that even though our intention was to provide enough details about our design to be able to cover both functional and non-functional requirements, we stopped short of selecting operating systems or product versions, since frequent upgrades there would soon render any book obsolete.

³⁰ For example, design patterns and coding guidelines

Important: This guide is essentially methodology-independent and can be woven into any existing industrial-strength methodology.

The methodology we have chosen is used by IBM Software architects to design solutions for a wide range of customer requirements. It includes three distinct phases—*Plan*, *Execute*, and *Implement*. Each of these phases has a well-defined set of tasks and activities that produce associated work products for helping the architect uncover and document system requirements and then to design a solution capable of meeting them. The *execute* phase is where reusable assets, best practices, and experiences can best be leveraged. Here requirements come to light and a solution is developed. We thread our step-by-step best practice process into the appropriate tasks and activities of this phase to show how reusable assets can be employed to seed work products and then customized to design a solution that meets unique customer requirements.

Note: These work products should be familiar to the architects and designers of modern e-business systems, and their notation is consistent with the Unified Modeling Language.

To better illustrate how to use this guide in actual practice, we have developed interrelated scenarios built around a complex customer case study. Here we chose a “typical” situation requiring modernization of disparate government agencies needing collaboration and integration. The business scenarios that comprise this study target the need for governments to collaborate across countries, agencies, regions, states, local governments, and the private sector in order to identify and more effectively control the sources of potential problems.

Table 3-1 on page 67 summarizes our best practice guide and sets the stage for the rest of the chapter. The first column shows the tasks, activities, and resulting work products in each phase of our solution development methodology . The second column shows the known solutions and assets that can be leveraged within a given part of that methodology, as well as the impact of reusing these assets. The final column shows the step-by-step best practice process that we recommend you follow to ensure that assets are leveraged effectively.

Note: The rows highlighted in yellow define our primary focus in writing this chapter.

Table 3-1 Best practice process & its relationship to development methodology & large reusable assets

Solution Development Methodology	Known Assets and Solutions	Best Practice Process for Leveraging Large Reusable Assets
Phase: Plan		
Activity: Evaluate Customer's Environment		
<p><i>Tasks:</i></p> <ul style="list-style-type: none"> ▶ Define Business Context ▶ Identify Business Issues and Goals 		
<p><i>Task:</i> Describe Current Organization</p>		
Activity: Develop Plan Linked to Customer's Business Initiatives		
<p><i>Task:</i> Document IT Standards</p>		
<p><i>Task:</i> Analyze Current IT Infrastructure</p>		
Phase: Execute		
Activity: Develop Customer Interest; Establish Buying Vision with the Customer	<p>Consider the catalog of IBM Patterns for e-business.</p> <p>Reference solutions, targeting the customers' industry and solutions that have been successfully developed for similar customers as potential sources of assets that can be leveraged for building the solutions to address the customers' needs.</p>	<p>Survey the catalog of large reusable assets, including the IBM Patterns for e-business, reference architectures, reference solutions targeting the customer's industry, and known solutions from customer engagements to build a set of candidate assets that can be leveraged to build your solution.</p>
<p><i>Task:</i> Obtain or Develop Business Roadmap</p>		

Solution Development Methodology	Known Assets and Solutions	Best Practice Process for Leveraging Large Reusable Assets
<p>Task: Gain Sponsorship</p> <p>Work Products:</p> <ul style="list-style-type: none"> ▶ Project Description ▶ System Context Diagram 		<p>Develop High Level Project Description and System Context Diagram</p>
<p>Activity: Demonstrate Business Benefits and Capabilities; Qualify Opportunity</p>		
<p>Task: Outline Solution Requirements</p> <p>Work Products:</p> <ul style="list-style-type: none"> ▶ Non-Functional Requirements (NFRs) ▶ Use Case Model 	<p>The reusable assets must be matched against the customer's requirements to validate that they can be effectively leveraged. When there is a match, the assets can be leveraged by showing which parts of these requirements are handled directly by the assets and the delta requirements that have to be custom-built.</p>	<p>Step 1: Develop the Solution Requirements:</p> <ul style="list-style-type: none"> ▶ <i>Step 1A:</i> List users/existing systems and functional requirements ▶ <i>Step 1B:</i> Derive actors and use cases; create Use Case Model ▶ <i>Step 1C:</i> Inventory large reusable assets (for example, Composite patterns, reference architectures, and industry solutions) ▶ <i>Step 1D:</i> Match solution use cases with generic use cases of the large reusable asset candidates to identify the matched large reusable asset. ▶ <i>Step 1E:</i> Identify delta use cases (additions or subtractions) to handle requirements that are not handled by matched large reusable assets and classify into business and integration patterns ▶ <i>Step 1F:</i> Identify and document NFRs ▶ <i>Step 1G:</i> Validate requirements with customer (for example, walkthrough storyboards).
<p>Task: Assess Initial Viability</p>		
<p>Activity: Develop Solution with Customer</p>		

Solution Development Methodology	Known Assets and Solutions	Best Practice Process for Leveraging Large Reusable Assets
<p>Task: Develop Solution Architecture</p> <p>Work Products:</p> <ul style="list-style-type: none"> ▶ Architecture Overview Diagram (AOD) ▶ Architecture Decisions. 	<p>Given the requirements gathered thus far, we want to develop the AOD. Known industry and customer solutions can also be leveraged by validating that the solution at least partially matches the requirements, and by using the known solution to seed the AOD. It is important to highlight which parts of the solution are not tackled by the asset, and then to work on designing and integrating these delta Business and Integration patterns to arrive at a complete solution</p>	<p>Step 2: Develop the Logical Design:</p> <ul style="list-style-type: none"> ▶ <i>Step 2A:</i> Derive Initial AOD (Subsystems), showing major functional subsystems by grouping use cases. ▶ <i>Step 2B:</i> Seed the AOD with matched large reusable asset (for example, Composite patterns, reference architectures, and industry solutions): <ul style="list-style-type: none"> – <i>Step 2B1:</i> Validate the AOD from matched large reusable asset matching requirements When using a Composite pattern, validate Application patterns. – <i>Step 2B2:</i> Capture chosen patterns into Architecture Decisions ▶ <i>Step 2C:</i> Identify and apply patterns for each AOD delta requirement <ul style="list-style-type: none"> – <i>Step 2C1:</i> Identify delta requirements that need to be addressed in the AOD – <i>Step 2C2:</i> Identify and apply Business and Integration patterns to address delta requirements – <i>Step 2C3:</i> Identify and apply Application patterns for delta Business and Integration patterns – <i>Step 2C4:</i> Capture chosen patterns into Architecture Decisions ▶ <i>Step 2D:</i> Transition the AOD from subsystem view to a logical node view <ul style="list-style-type: none"> – <i>Step 2D1:</i> Seed the AOD (Logical Nodes) with logical system architecture from matched large reusable asset (for example, the Composite pattern's Runtime patterns)

Solution Development Methodology	Known Assets and Solutions	Best Practice Process for Leveraging Large Reusable Assets
		<ul style="list-style-type: none"> – <i>Step 2D2:</i> Identify and apply Runtime patterns for the delta Application patterns – <i>Step 2D3:</i> Integrate Runtime pattern for the Application patterns into the AOD – <i>Step 2D4:</i> Capture chosen patterns into Architecture Decisions ▶ <i>Step 2E:</i> Review and apply logical design guidelines to the AOD (Logical Nodes) ▶ <i>Step 2F:</i> Conduct technical walkthroughs to ensure that functional requirements are met
Conduct technical walkthrough		

Solution Development Methodology	Known Assets and Solutions	Best Practice Process for Leveraging Large Reusable Assets
<p><i>Task:</i> Develop High-level Component Model (Optional)</p> <p><i>Work Product:</i> Component Model</p>	<p>The Component Model includes both functional and technical component interaction models. Existing Patterns for e-business redbooks might help in the development of these Component Models. For the identified Business, Integration, and Application patterns, select the appropriate Patterns redbook and review its design guidelines chapter. This chapter will outline the technical component interactions for a given solution, based on the patterns identified earlier in this process. The information gathered from the design guideline chapters in the Patterns for e-business redbooks should be used as a baseline for component interactions. The component model for known solutions and the established best practice guidelines for the type of solution can be leveraged to seed the Component Model.</p>	<p>Step 3: Develop the Physical Design:</p> <ul style="list-style-type: none"> ▶ <i>Step 3A [optional]:</i> Build the Component Model, showing software component nodes (required to support functional subsystems). ▶ <i>Step 3B [optional]:</i> Apply design guidelines (application level) from matched large reusable assets, including redbooks and best practice white papers. ▶ <i>Step 3C [optional]:</i> Conduct technical walkthrough of component interactions

Conduct technical walkthrough

Solution Development Methodology	Known Assets and Solutions	Best Practice Process for Leveraging Large Reusable Assets
<p>Task: Develop Operational Model</p> <p>Work Product: Operational Model</p>	<p>When leveraging patterns:</p> <ul style="list-style-type: none"> ▶ Finalize the Application patterns to be used, again based on the Business, Integration, and Composite patterns selected earlier. Here you can select the Runtime pattern for each of the identified Application patterns based on Non-Functional Requirements (NFRs). ▶ If multiple Runtime patterns are selected, combine to get a first draft of the Operational Model. You might need to develop several runtime architectural options. <p>When leveraging known solutions:</p> <ul style="list-style-type: none"> ▶ Use runtime product mappings that are known to work together. ▶ Leverage scalable architecture techniques to apply NFRs to the operational model, especially to address high availability. <p>Leverage known operational configurations to address system management issues.</p> <p>Note: The Component and Operational Model tasks can be developed simultaneously. In some cases, the information from the Operational Model will be needed to evaluate the technical component interactions in the Component Model.</p>	<ul style="list-style-type: none"> ▶ <i>Step 3D:</i> Seed the Operational Model by applying product mappings from the matched large reusable asset. Apply product mappings for the delta requirements. ▶ <i>Step 3E:</i> Address “operational” NFRs in the Operational Model ▶ <i>Step 3F:</i> Conduct technical walkthroughs to ensure that the NFRs are met

Solution Development Methodology	Known Assets and Solutions	Best Practice Process for Leveraging Large Reusable Assets
	As such, the architect or individual completing these tasks in the method should not assume that the Component Model must be completed before beginning the Operational Model.	
<i>Task:</i> Refine Viability Assessment		
	Conduct technical walkthrough	
Activity: Refine Solution, Resolve Concerns, Close Sale		
<i>Task:</i> Assess Business Impact		
<i>Task:</i> Ensure Client Commitment		
<i>Task:</i> Evaluate Integrated Solution		
Phase: Implement		
Activity: Monitor Solution Implementation and Ensure That Expectations Are Met		
<i>Task:</i> Monitor Pilot		
<i>Task:</i> Evaluate Success		
<i>Task:</i> Harvest Assets		
Note: All of the best practices are primarily applicable to the <i>Execute</i> phase and support the development of the work products used to model the solution. Figure 3-1 on page 74 illustrates the dependencies between the primary work products.		

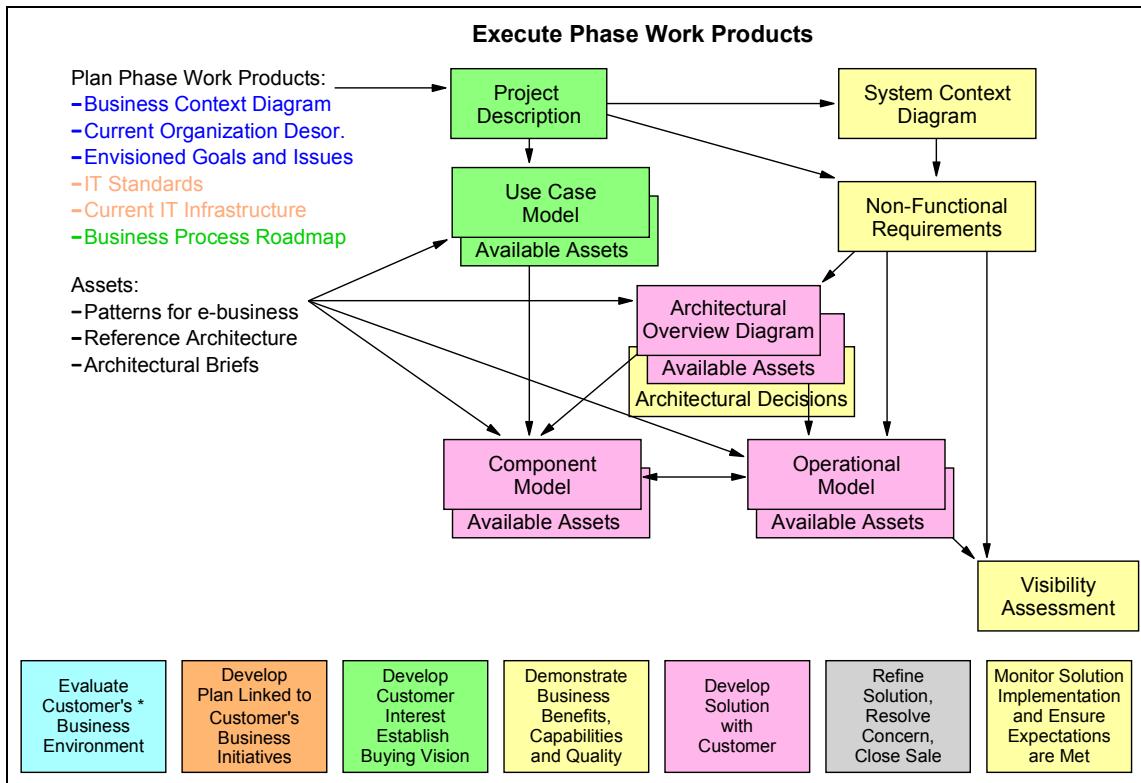


Figure 3-1 Work product dependencies

As promised, some of these work products will be examined within this chapter to assist you in understanding how they contribute to the design of your solution.

Note: We have assumed that the reader is familiar with work products and the Patterns for e-business. This redbook is not intended to cover those patterns or work products other than to show how they are used in architecting a portal solution. For further information on the IBM Patterns for e-business, go to <http://www.ibm.com/developerworks/patterns/>.

Since this redbook is about developing a portal solution, we have focused on the Portal composite pattern. The next section consists of a “sidebar” providing further details about the Portal composite pattern. Readers familiar with this can skip to 3.1, “Plan” on page 77.



Sidebar: Portal composite pattern

This pattern is designed to facilitate many variations of similar functionality. A portal solution typically aggregates multiple information sources and applications to provide a single, seamless, personalized access to users.

As shown in Figure 3-2, the Composite pattern for portal applications is made up of an Access Integration pattern that facilitates functions such as single sign-on, multiple device support, and personalization—and integrates the common Business patterns for portal applications, including Self-Service, Information Aggregation, and Collaboration.

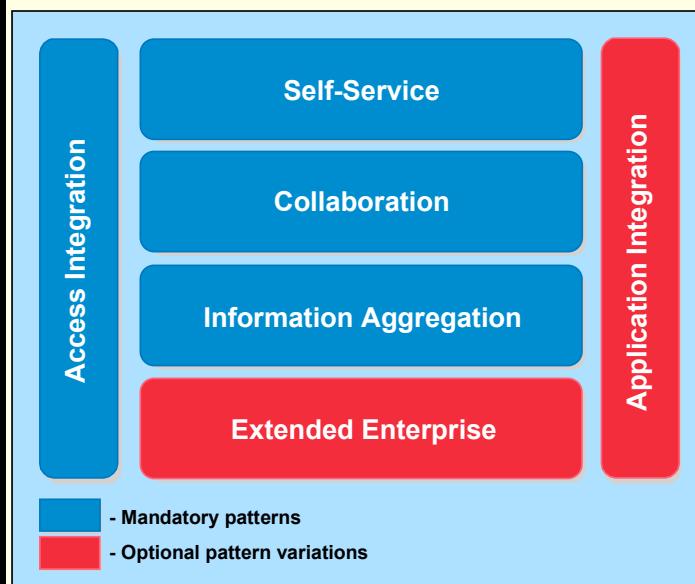


Figure 3-2 Portal composite pattern

Benefits

Once an organization has determined that it needs to aggregate information, target that information to specific users, analyze the usage of information, and collect and manage information, it can use a portal to meet these requirements. Creating a portal architecture can provide the following benefits:

- ▶ A single aggregated view of content targeted to specific user types
- ▶ Ability to analyze usage patterns to make marketing efforts more efficient

- ▶ Ability to tailor the user interface to specific groups, thereby enabling a focus on cultural, language, or nationality-based differences
- ▶ Single sign-on, allowing the user to save time and have access to information while decreasing the requirements for direct interaction with the organization, thereby saving money

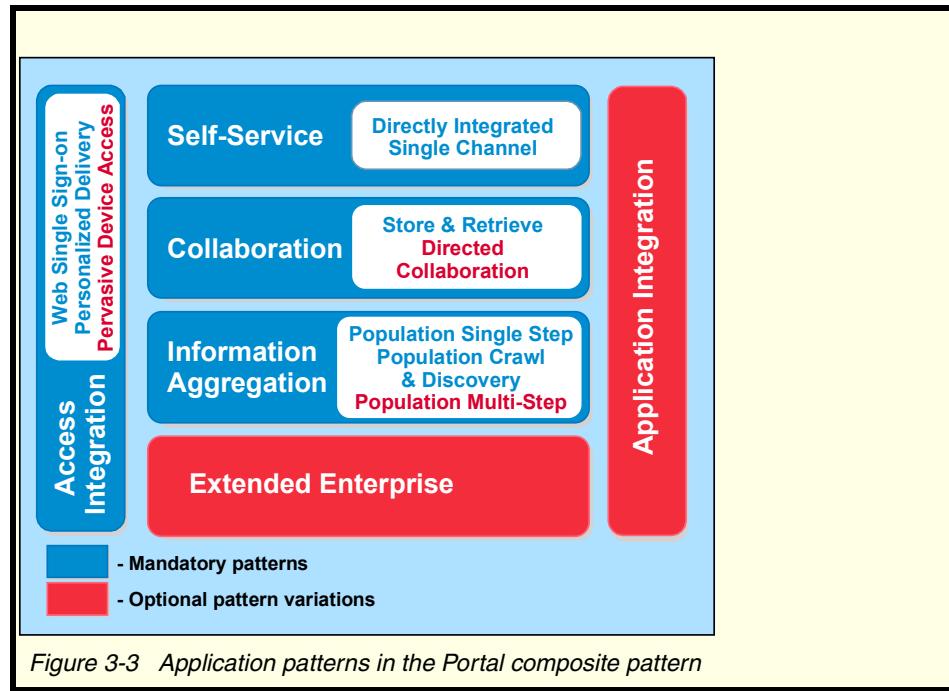
Application patterns of the Portal composite pattern

An Application pattern shows the principal layout of the application, focusing on its shape, logic, and associated data. It identifies the high-level logical components needed to implement the key functions of the chosen Business patterns.

The selection of an Application pattern is based on the selected Business, Integration, and Composite patterns. Application patterns use logical tiers to illustrate the various ways to configure the interaction between users, applications, and data.

Each Application pattern has associated business and IT drivers. Review each of the business and IT drivers with the associated Application pattern to determine the best fit for the requirements.

Figure 3-3 on page 77 shows the specific Application patterns (within the associated Business and Integration patterns) that can be used to enable the various types of functionality found in each Business and Integration pattern. Note that some of these Application patterns are mandatory (shown in blue text) for a properly functioning portal, while others are optional (indicated by red text).



3.1 Plan

The purpose of this phase is to formulate and document an overall solution plan. The plan should clearly link to the customer's business strategy and initiatives. For example, using a portal solution plays into a company's overall e-business strategy. The plan should detail the scope of the project (the customer's envisioned goals and the business context—in other words, internal and external relationships).

Note: Best practices do not provide any leverage in the *Plan* phase. So, for this phase we will focus on providing case study details concerning envisioned goals and issues.

Task: Document envisioned goals and issues

Part of the issues discovery process is determining the customer's current problems and the direction in which it sees itself moving. Documenting this is important because it helps determine high-level functional requirements, which in turn help scope the project and make good architectural decisions. As part of this

process, you should investigate the company's mission and vision statements along with future business goals.

For our case study, these goals are as follows:

- ▶ To increase the effectiveness and timeliness of any government's security, a new infrastructure must be designed to allow disparate government agencies, using information technology, to more effectively share information and intelligence in order to respond to any potential threat. This needs to happen both horizontally (among Federal agencies and Departments) and vertically (among the Federal, state, and local governments). This will allow Federal, state and local government agencies, as well as the private sector, to work together seamlessly. Having the right system of communication (content, process, and infrastructure) is critical to bridging the existing gaps between these agencies and organizations.
- ▶ The future goals or vision of e-government is all about improving citizen services, creating efficiencies to reduce costs, and empowering citizens and businesses that are demanding better and more efficient services. Drivers for this vision range from legislative mandates to market forces such as globalization, supply chain integration, trans-national crime and terrorism, environmental and social threats, and technology trends such as pervasive wireless and advanced integration and analysis techniques.

Figure 3-4 on page 79 summarizes e-government characteristics and key initiatives.

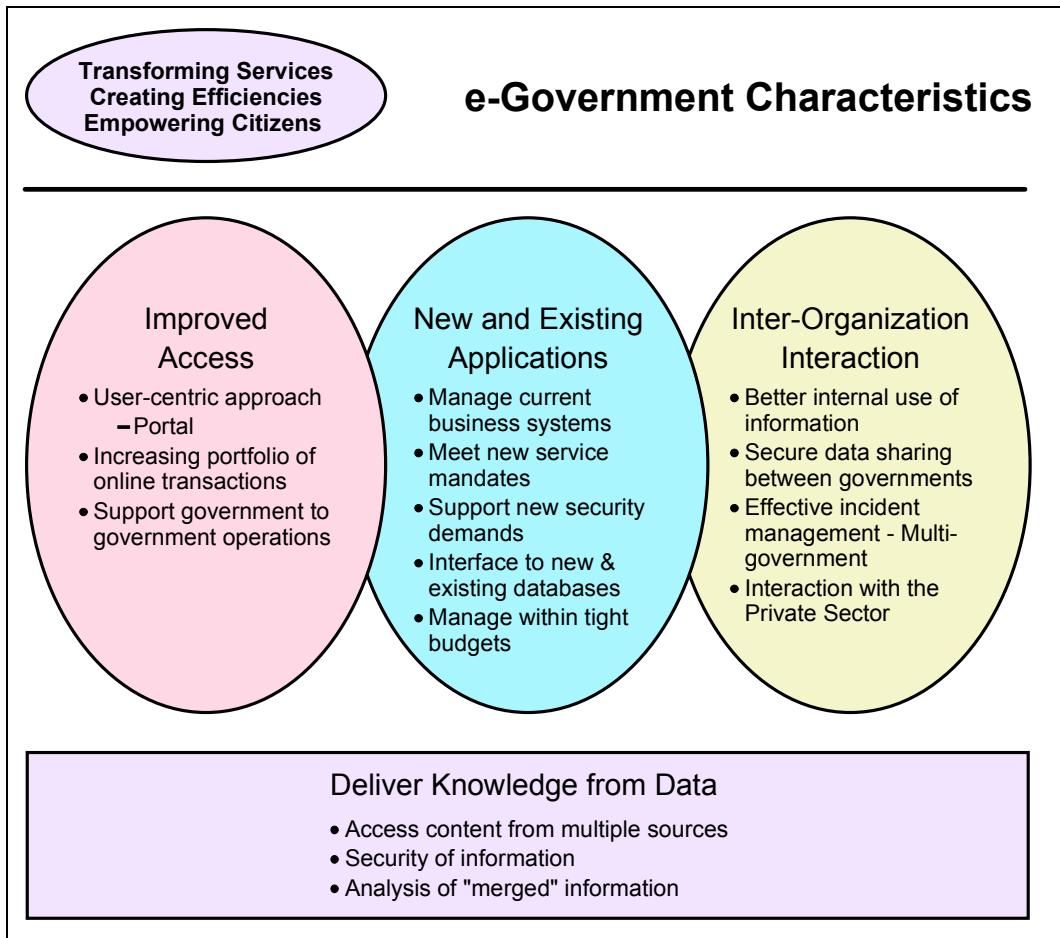


Figure 3-4 e-Government characteristics and key initiatives

The following describe these key initiatives for e-government:

► Improved access

Improving access includes enabling transactions for citizens, agencies, and the private sector by whatever channel they choose to utilize and providing the flexibility to adapt to new technologies as they become available

► New and existing applications

Integration is essential to running a government in today's world, requiring both process and technical integration. To meet these challenges, government agencies must look to delivering more efficient services while managing their budgets. Taking advantage of new technologies that enable more self-help capability can provide a means to deliver services and

information to citizens and businesses while reducing the overhead of more expensive resources.

Leveraging existing systems supports the visions of e-government by the cost-effective measure of re-using systems that are already in place and proven to be reliable and functional. Transformation of existing processes and effectively integrating to existing systems are crucial to providing new levels of services.

- ▶ Inter-organization interaction

Citizens and businesses need to interact with all levels of government. Therefore, to realize the vision of e-government, intergovernmental linkages must be developed and facilitated through underlying technology. Providing better communications capabilities that use interactive messaging facilities, Web portals that aggregate data from disparate systems, electronic team rooms, and messaging gateways to facilitate information exchange could all be established as a way to enable cross-agency, cross-government, and government-to-private-sector information sharing.

Collaboration can empower businesses and citizens while providing better governmental services.

- ▶ Deliver knowledge from data

It is imperative that you obtain data from various sources, aggregate and sort this data, and present meaningful results from it.

In essence, the future business goals are to provide for better communication and information sharing across disparate agencies. It plans to allow government organizations and private individuals to work together more efficiently and effectively. The use of existing systems integrated with new systems is the infrastructure to enable these goals.

Note: The output of this task is the Envisioned Goals and Issues work product.

3.2 Execute

The purpose of this phase is to create an architecture that translates business requirements into a solution. As with any software application, the solution architecture should address both the functional and operational domains.

The *functional domain* addresses the business functionality of the solution. It describes the structure and modularity of the software components that will be used to realize the business functions, the interactions between the components, and the interfaces between them.

The *operational domain* describes the system organization (such as hardware platforms, connections, locations, and topology), the placement of software and data components, the NFRs (such as performance, availability, and security), and system management (capacity planning, software distribution, and backup and recovery).

Remember that the architecture needs to align with the overall business strategy and provide a roadmap for its implementation.

The Execute phase begins with the establishment of a project within which a solution will be developed. There are three steps in designing the solution:

1. Establish system requirements through defined use cases.
2. Design a solution that meets these requirements. The logical design of this system is defined primarily through an *Architecture Overview Diagram*. The associated key decisions are captured in the *Architecture Decisions* work product.
3. The *logical design* is transformed into an operational physical design by defining the components that make up the logical system and the packaging of these components into applications that run on a physical infrastructure of servers, storage, and networking. The *physical design* is primarily modeled using the Component and Operational Models.

Task: Develop high-level Project Description and System Context Diagram

The *Project Description* establishes the scope of the project. It describes what is to be accomplished and identifies the high-level functions that must be taken into account by the solution. The *Envisioned Goals and Issues* document should be used as input to the Project Description. You need to know where the customer is heading and what is its vision for the long term. Otherwise, you might be creating a solution that is valid only for the short term. It's also important to list the business and IT drivers for the project. These will be used to drive and validate your choices during the logical and physical design steps.

For our e-government case study, the Project Description is as follows:

This Incident Management project addresses the need for government agencies to work together to manage low security threat incidents (for example, a contamination of a food shipment). Agencies that participate in some aspect of the incident have information pertinent to that incident and subject matter experts who can resolve it. These experts are required to participate in a collaborative environment, working together to resolve the incident effectively and efficiently. The following define the major functional areas for managing the incidents:

- ▶ Function—Vehicle/Vessel Enters Port
 - Vessel requests Port Authority permission to dock, sending manifest and crew list
 - Assess risk by integrating information from multiple agencies
 - Trigger Customs inspection (if needed)
- ▶ Function—Proliferation and Tracking
 - Determine if cargo may be contaminated
 - Alert subscribing government agencies
 - Correlate information across agencies to track extent of problem and assign lead agency
- ▶ Function—Collaboration
 - Identify experts and create cross-agency incident response team
 - Set up collaborative environment for team
- ▶ Function—Interception/Treatment
 - Disseminate information to regional emergency teams and businesses
 - Determine public communication strategy
 - Approve and publish bulletins

The following scenarios show a typical end-to-end use of the Incident Management System functionality described above:

- ▶ ***Scenario 1—Vessel Enters Port.*** A large foreign commercial container ship submits a request to the Coast Guard “Master of the Port” for permission to dock at the port. The request includes a manifest and crew list. A risk assessment is performed on the ship's cargo and the crew list, integrating information results from across agencies. The risk factor is determined to be high, and it is determined that a close inspection of the ship's cargo is required.
- ▶ ***Scenario 2—Proliferation and Tracking.*** The Customs Agency is called in to perform the inspection, during which the Inspector finds the contents of one of the containers to be contaminated. Due to the serious nature of the inspection failure, an alert is sent to interested (subscribing) government organizations.
- ▶ ***Scenario 3—Collaboration.*** An emergency response team is created, composed of selected experts. A collaborative environment for the team is set up. Intensive analysis is performed on data in a variety of agencies to find relevant information for this incident. Investigation determines that additional shipments with the same lot number were scheduled to enter the country from other ports around the country. The Department of Agriculture and the US

Customs Service discover that some shipments have cleared customs and are either in transit to grocery stores or already on the shelves. It is decided that a Health Alert Bulletin must be created and disseminated immediately.

- ▶ **Scenario 4—Interception and Treatment.** A document is created on the CDC (or other associated response team's) Alert Content Management/Publishing System and published to the inter-agency Alert System. The document is then published and disseminated to the public, where a person who works in the food industry sees the news flash on her company's portal. She realizes that her company sells this product and immediately informs her store managers and warehouses to pull the product from the shelves, thereby preventing an occurrence of someone becoming ill from consuming a contaminated substance.

Business drivers for the case study

- ▶ Need to integrate “stovepipes” and decrease redundancy
- ▶ Need to streamline processes and improve data analysis
- ▶ Need to facilitate effective decision-making

IT drivers for the case study

- ▶ Better internal use of information
- ▶ Secure data sharing
- ▶ Effective incident management through automated processes
- ▶ Support for new security demands
- ▶ Interface to new and existing databases

The *System Context Diagram* is intended to exhibit the relationships, interactions, and information flow into and out of enterprise systems. These diagrams also serve as a discussion aid in determining project scope. Figure 3-5 on page 84 represents the System Context Diagram for our case study. This diagram shows the disparate government organizations, the customers, and the outside sources of information, coupled with the relationships that comprise the solution. Any interaction between the agencies is via telephone, fax, etc. The *vision* is to enable and enhance electronic communication and collaboration.

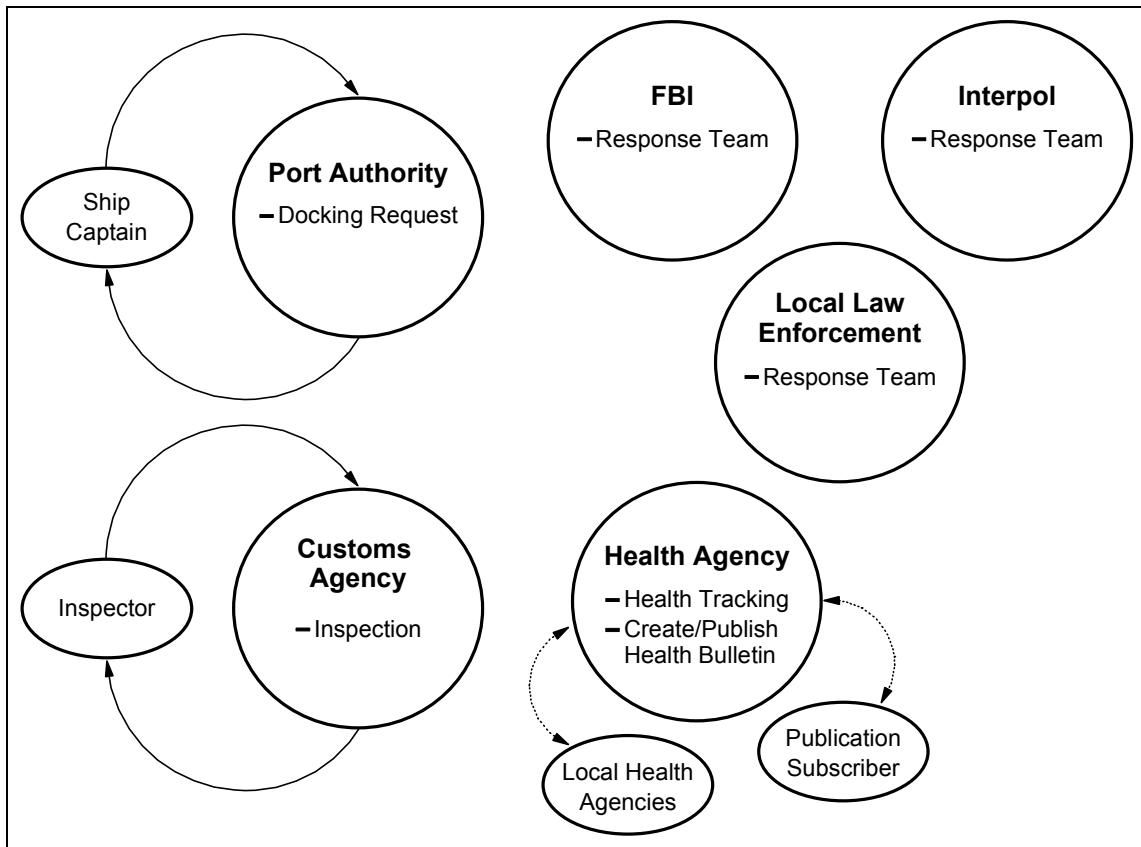


Figure 3-5 System Context Diagram for e-government

Note: The outputs of this task are the *Project Description* and *System Context Diagram* work products.

3.2.1 Step 1: Develop solution requirements

The goal of this step is to define the solution requirements and to understand their complexity and scope.

The process of outlining solution requirements involves identifying both the *Functional* and *Non-Functional Requirements* and then documenting the architectural principles that will drive the solution. The process begins by creating a description of the project in order to develop the solution outlining the major functionality that is needed. A *System Context Diagram* shows the current system environment in which the solution will exist. Using the *Project Description*

and *Current System Context*, the Functional Requirements are derived and documented in a *Use Case Model*.

The following reiterates the best practice process for leveraging large reusable assets to develop solution requirements:

- ▶ Step 1A: List users/existing systems and Functional Requirements
- ▶ Step 1B: Derive actors and use cases required to create the Use Case Model
- ▶ Step 1C: Inventory large reusable assets
- ▶ Step 1D: Match solution use cases with generic use cases of the large reusable asset candidates
- ▶ Step 1E: Identify delta use cases to handle requirements that are not handled by the matched large reusable asset and classify into Business and Integration patterns
- ▶ Step 1F: Document Non-Functional Requirements
- ▶ Step 1G: Validate requirements with customer

Each of these process steps will be explained in its respective section below.

Step 1A: List users/existing systems and Functional Requirements

From the *High-Level Description* and *System Context Diagram* we will glean the actors, high-level business functions, and high-level interactions between these actors and functions. The first step in this identification process is to list the users and existing systems from these work products and to identify the major functionality required by the system.

Based on the Project Description and the System Context Diagram, we derive the following list of actors and existing systems as well as the major functionality needed:

- ▶ Users and existing systems:
 - Vehicle/vessel/ship
 - Captain
 - Agency
 - Port Authority
 - Customs
 - Inspector
 - Emergency Management (Homeland Security)
 - Agent
 - Health agency
 - Incident response team

- Lead agency
- ▶ Functionality required:
 - Request permission to dock
 - Assess risk
 - Integrate information from multiple agencies
 - Trigger inspection
 - Inspect manifest
 - Alert agencies
 - Assess problem scope
 - Assign lead agency
 - Identify experts
 - Create incident response team
 - Setup collaborative team environment
 - Disseminate information
 - Approve and publish bulletins

Step 1B: Derive actors and use cases required to create the Use Case Model

A *Use Case* describes the flow or sequence of events when an actor³¹ interacts with the system, and is used to establish the most important scenarios for using the application or system under consideration. It helps determine the required functionality and user interfaces. Use cases don't describe how the system works internally. Figure 3-6 on page 87 depicts the use cases that were derived from our list of users and existing systems as well as the functionality required

³¹ An actor can be a class of users, roles users play, or other systems.

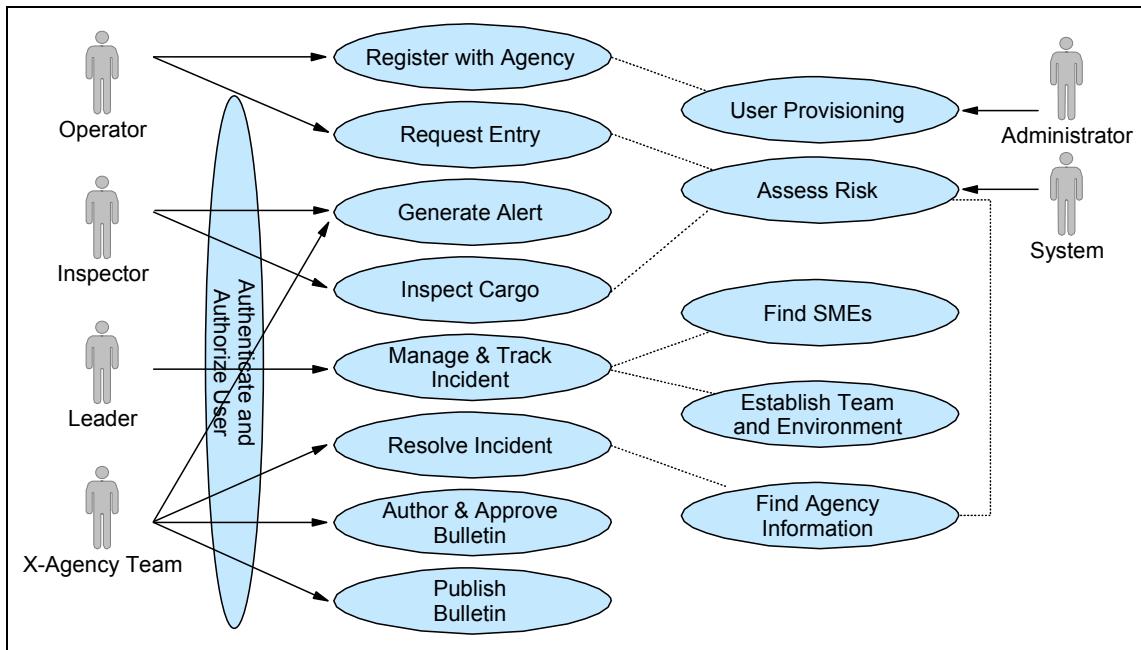


Figure 3-6 e-Government Use Case Model

Table 3-2 provides short descriptions of each use case:

Table 3-2 e-Government Use Case descriptions

Use Case	Description
Authenticate and Authorize User	All users authenticate with the system, establishing a distinct role for interaction with it.
Author and Approve Bulletins	The X-Agency incident management team puts together a bulletin to handle the incident, which must be approved to be published. For example, the incident management team creates a health alert bulletin explaining that there is a product recall on a contaminated item.
Resolve Incident	The X-Agency team—comprising experts from various agencies—performs tasks, shares information, and collaborates to manage the incident.
Inspect Cargo	The inspector checks the cargo for potential hazardous materials and contamination. Usually, this individual uses a handheld device to enter inspection results.

Use Case	Description
Publish Bulletin	The X-Agency team publishes an approved bulletin as the result of managing the incident.
Manage and Track Incident	The incident leader establishes an X-Agency team of experts to resolve the incident. To facilitate the team, the leader establishes a collaborative environment for it and populates this environment with relevant information from various sources/agencies. The leader also tracks the incident to ensure that it is being handled effectively and in a timely fashion.
Request Entry	The operator of a vehicle or vessel requests entry into a port. Typically, the request for entry requires a crew list and manifest information.
Register with Agency	Operators of vehicles or vessels register with the agency to use its services.
User Provisioning	The system administrator manages user accounts, including assigning roles.
Assess Risk	The system assesses the risk of a request through an established process that uses each agency's information and services to determine any potential problems.
Generate Alert	The system generates an alert based on its assessment of the request. Typically the alert is an indicator for an inspection.
Generate Incident	An incident is generated when an inspection determines that a problem exists.
Find SMEs	An X-Agency team requires the identification of experts in different fields to tackle an incident. Finding subject matter experts facilitates the generation of a list of potential candidates from the various agencies.
Find Agency Information	Each agency has information that could be pertinent to handling the incident and assessing risk. Finding agency information is required to facilitate the X-Agency team to resolve the incident.
Establish Team and Environment	An integral part of setting up a X-Agency team is to establish the team and collaborative environment.

Note: The output of this task is the Use Case Model work product.

Step 1C: Inventory large reusable assets ³²

Here we determine if there are any large reusable asset candidates—including Composite patterns for e-business, reference architectures, industry solutions, and other solution architectures, implementations, engagements, etc. If so, we investigate them to see how close the business requirements match. To be useful, these large reusable assets must have a complete set of work products that can be leveraged through the best practice process. For this step, the large reusable asset must describe the business and IT drivers and a set of generic use cases.

Since by matching the needed functionality to the characteristics of portal solutions outlined in Chapter 2, “Portal solutions” on page 39 we know that the intended project is a portal solution, we can consider leveraging the Portal composite pattern. The business and IT drivers for the Portal composite pattern shown in Figure 3-7 on page 90 indicates a potential match to the intended project:

Business drivers

- ▶ Deep customer knowledge and mindshare
- ▶ Achieve leadership from a quality and/or marketplace mindshare perspective
- ▶ Transactional and process efficiency

IT drivers

- ▶ Information and data aggregation
- ▶ Information targeted to specific user and group types
- ▶ Identification of user and group types
- ▶ Content management (including workflow)
- ▶ Search and indexing of static and dynamic database content
- ▶ Single sign-on
- ▶ Centralized business rules definition
- ▶ Centralized security mechanism (covering single sign-on functionality)

The generic Use Case Model for the Portal composite pattern (see Figure 3-7 on page 90) highlights the Business and Integration patterns associated with the use cases.

³² For example, Composite patterns, reference architectures, and industry solutions

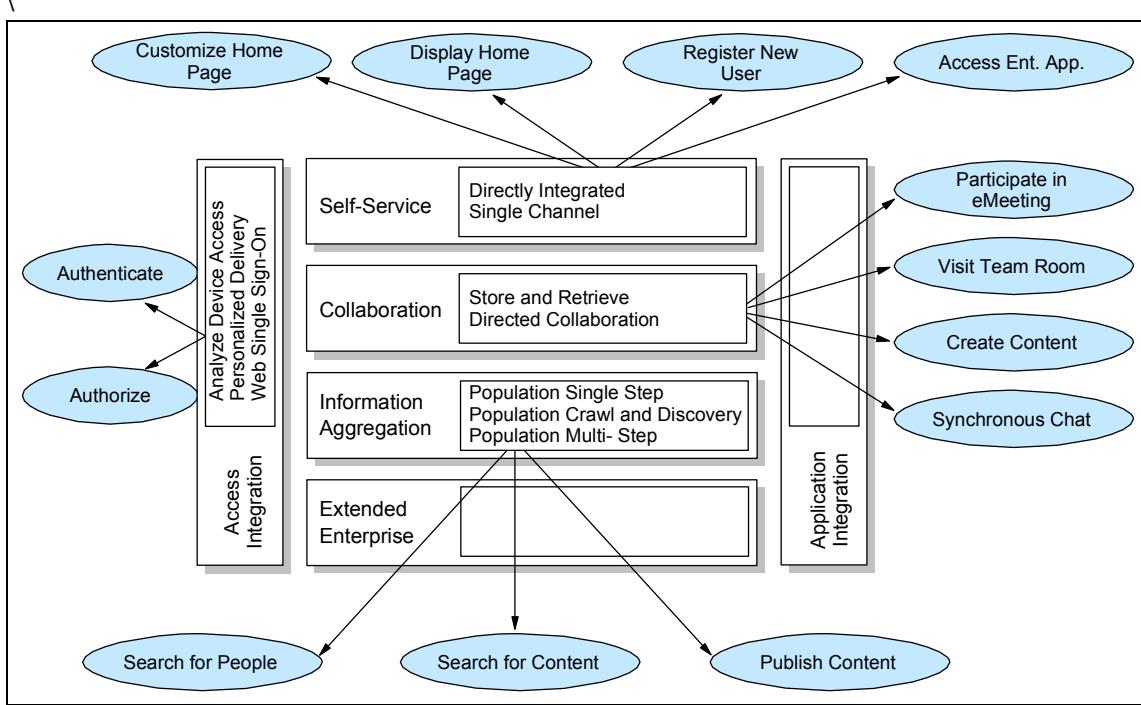


Figure 3-7 Portal composite pattern generic use cases

Note: The Use Case Model above matches very well with the requirements, making the Portal composite pattern a strong asset candidate that can be leveraged for the solution.

Step 1D: Match solution use cases with generic use cases of the large reusable asset candidates

Now that you have identified asset candidates to leverage, compare the intended solution use cases to the generic use cases of the large reusable assets. In this section, we will focus on matching the intended solution's use cases to the generic use cases of the Portal composite pattern.

Table 3-3 on page 91 describes these Portal composite use cases.

Table 3-3 Portal composite pattern generic use case descriptions

Generic Use Case	Description
Register New User	Registering users to the portal system
Authenticate	Validating the user to the portal system
Authorize	Determining the user's access privileges once authenticated
Customize Home Page	Allowing users to personalize their home page content
Display Home Page	Displaying a home page based on the user's role, displaying only the information pertinent to that role
Access Enterprise Applications	The system accesses an enterprise application, providing data and functionality
Visit TeamRoom	Providing an asynchronous area for user collaboration
Create Content	Allowing users to author, edit, and (through workflow with other users) collaborate on content from within the portal system
Synchronous Chat	Allowing users to send instant messages to other online users
Participate in e-meeting	Facilitating real-time communication for user access to various types of media content (for example, presentation files, applications, or streaming media) through the portal
Search for People	Providing the ability to search for other portal system users
Search for Content	Providing the ability to search for various content in the portal system
Publish Content	From the portal interface, allowing users to publish content to the portal system

Matching the intended solution to the Portal composite pattern's use cases yields the mapping depicted by Table 3-4.

Table 3-4 Solution use cases that match Portal composite pattern generic use cases

Solution Use Cases	Matched PCP Use Cases
Register with Agency	Register new user (SS)
Authorize and Authenticate User	<ul style="list-style-type: none"> ▶ Authenticate (Accl) ▶ Authorize (Accl)
Establish Team and Environment	Visit TeamRoom (C)

Solution Use Cases	Matched PCP Use Cases
Manage and Track Incident and Resolve Incident	<ul style="list-style-type: none"> ▶ Create content (C) ▶ Synchronous chat (C) ▶ Participate in e-meeting (C)
Find SMEs	Search for people (IA)
Request Entry	Access remote applications (SS)
Author and Approve Bulletin	Content creation and management (C)
Publish Bulletin	Publish content (IA)

Note: We have indicated to which Business (SS = Self Service, C = Collaboration, IA = Information Aggregation) or Integration (Accl = Access Integration) pattern the use case belongs.

Step 1E: Identify delta use cases to handle requirements that are not handled by the matched large reusable asset and classify into Business and Integration patterns

The following use cases do not directly match any generic use cases associated with the Portal composite pattern:

- ▶ Generate Alert
- ▶ User Provisioning
- ▶ Inspect Cargo
- ▶ Assess Risk
- ▶ Find Agency Information ³³

Next, we classify the remaining use cases into business and integration patterns to show how these “deltas” will be integrated into the initial solution that can be leveraged from the reusable asset.

Figure 3-8 on page 94 maps the remaining use cases into the primary Business patterns, and (where appropriate) to an Integration pattern that could potentially be used to integrate the functionality into the baseline solution:

³³ Part of this use case matches *Search for content*. However, this doesn't take into consideration the external systems that must be accessed as part of this search.

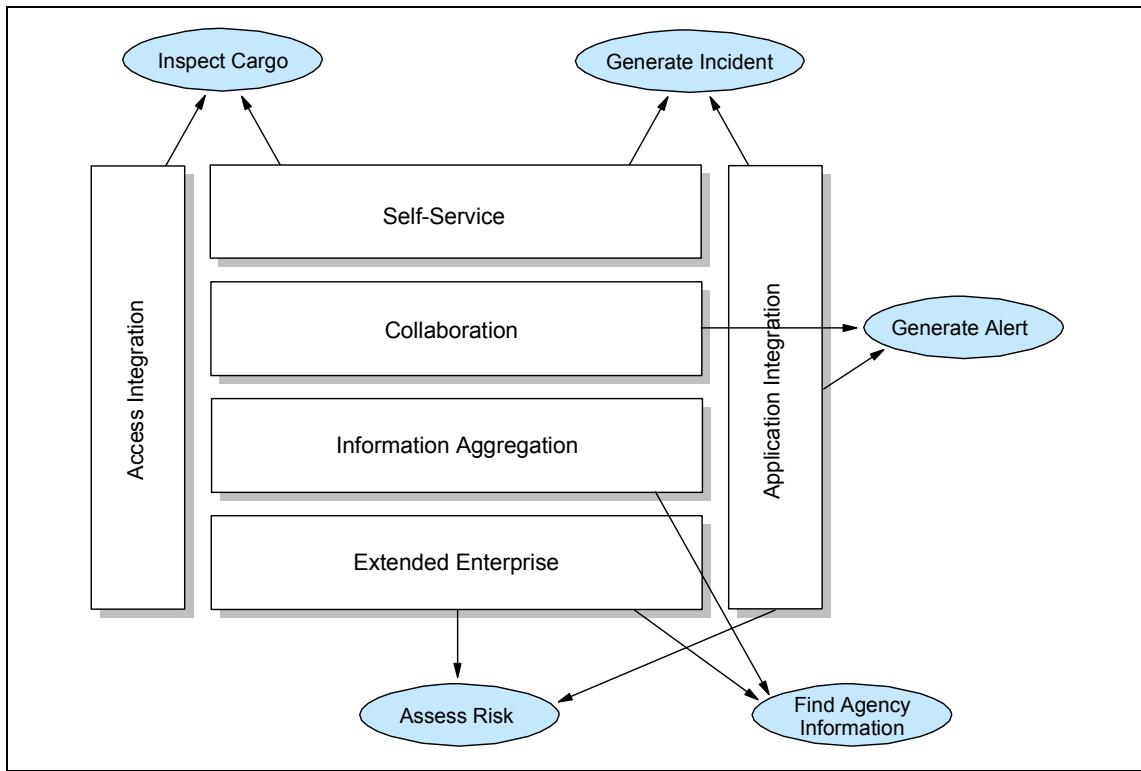


Figure 3-8 Classifying delta use cases

Looking at the details, each of Generate Alert (Apia), Generate Incident (Apia), Find Agency Information (EE), and Assess Risk (EE) include a Business or Integration pattern that is not part of the Portal composite pattern. *Inspect Cargo* includes an optional Application pattern for Access Integration (Pervasive Device Access).

Step 1F: Document Non-Functional Requirements

Non-Functional Requirements (NFRs) are those technically oriented requirements against an IT infrastructure that are necessary to support the business-oriented requirements but oriented to the IT community only. Since NFRs indirectly support use cases through functional requirements, they are closely related to the Use Case work product addressed previously in this chapter (see Step 1B: Derive actors and use cases required to create the Use Case Model). NFRs demand IT infrastructure function as opposed to application function, and define what the system must do from an infrastructure viewpoint. Since they frequently define key features, characteristics, and constraints on the proposed IT system, NFRs are among the *most important requirements affecting*

the ultimate architectural decisions. The easiest way to remember them is that they represent the “ilities”—usability, manageability, scalability, portability, reliability, availability, and maintainability. Other key items addressed by NFRs are *performance and security*. They are frequently the first sign of potential software product requirements.

NFRs are used as a basis for early system sizing and cost estimates, as well as to assess the viability of the proposed IT system and to define requirements and constraints upon it. Clear requirements are necessary for a successful project because they define the project's goals, clarifying what is needed in the end solution. They also *drive the design of the operational models, architectural decisions (since they frequently define key features, characteristics and constraints on the proposed system), and component design*. Many system constraints having significant impacts on the ultimate system architecture are uncovered while discovering the NFRs. Constraints might include the use of specific hardware or software, which in themselves may have associated constraints such as operating system prerequisites or release levels. There may be physical constraints such as the physical location of hardware, technology or standards constraints (such as the use of Java or Linux), operational constraints (such as wireless capability), or configuration constraints (such as using an existing network infrastructure). Non-Functional Requirements are frequently the most important determining factor of the architecture. Two systems with the same use cases but with very different Non-Functional Requirements need very different architectures.

The NFRs for the e-government case study follow.:.

Performance

- ▶ Concurrent accesses from multiple users to be supported
- ▶ Expect 20,000 agency users plus the external users registered with agencies

Availability

- ▶ Configurable to achieve 24x7 but not required in the first installation
- ▶ Data access limited by legacy system availability

Maintainability

- ▶ Variable data should be maintained as metadata in properties files so that changes can be made without recoding.
- ▶ Presentation, business logic, and data abstraction must be enforced to reduce the potential impact of changes.

Security

- ▶ Authentication—The following to be provided:
 - Access validation by user ID/password

- Usage of passwords for supervisor functions
- System administrator authority to set up and maintain application security for functions and data
- An interface to authenticate clients (for use by server software)
- ▶ Authorization:
 - Validation of authorized clients to be provided by server components
- ▶ Other:
 - Passwords to be masked (not printed) at entry
 - Passwords to be stored in encrypted format
 - PKI or Basic authentication/authorization
 - LDAP-based

Manageability

- ▶ Change control information:
 - Change control for this solution should adhere to current practices within the agency.
- ▶ System administration—The following to be provided:
 - The ability to start and stop software components
 - The ability to display alarms and events from software components
 - The ability to set up and maintain user profiles (for example, roles)
- ▶ Installation and distribution—The following to be provided:
 - Installation of client and server software components
 - The ability to distribute/push files
 - Following installation:
 - Procedures to perform system verification
 - Tools to perform system verification
 - Support—both on demand and by scheduled distribution

System Usability

- ▶ User interface designed for non-technical users
- ▶ Robust capabilities to execute business rules during data entry and transaction processing
- ▶ Strategic trend toward integrated business operations, driving requirements for back-end enterprise data integration
- ▶ Strategic trend toward e-government services, driving tactical trend toward front-end Web enablement or enterprise portal with CRM functionality and e-services models

Data Integrity

- ▶ Delivery of messages regardless of the state of the network or applications

- Logic for retry and system alert if problems exist in completing a transaction

Note: The output of this task is the NFRs work product.

Step 1G: Validate requirements with customer³⁴

The final task is to validate the requirements with the users of the systems, usually by walking through the intended functionality using storyboards. Since this task is difficult to show in a book format, we will skip this step in the narrative.

3.2.2 Step 2: Develop the logical design

The logical system design is driven by the Architecture Overview Diagram (AOD), which takes into account the NFRs, the architectural decisions, and the IBM Patterns for e-business Composite, Business, Integration, and Application pattern mappings. From these, the Runtime patterns are derived.

We already have the NFRs and the Architectural Decisions document from the step we just completed (see 3.2.1, “Step 1: Develop solution requirements” on page 84). We will continue the formulation of the logical solution by creating the AOD.

The AOD will go through two stages.

1. The first will identify the subsystems of the solution. We will refer to this as the *Architecture Overview Diagram (Subsystem)*. This version of the AOD is most compatible with the notation used in working with Business, Integration, Composite, and Application patterns.
2. The second will denote the logical nodes within the solution. We will call this the *Architecture Overview Diagram (Logical Nodes)*. This logical nodes version of the AOD is aligned with the notation used by the Runtime patterns and is compatible with the typical work product used by most architects.

Steps required to develop the logical design

- Step 2A: Derive the initial Architecture Overview Diagram
- Step 2B: Seed the AOD with large reusable assets
- Step 2B1: Validate the AOD from matched large reusable asset matches requirements

Note: When using a Composite pattern, validate Application patterns.

³⁴ For example, walkthrough storyboard(s)

- Step 2B2: Capture decisions for chosen patterns into the Architecture Decisions document
- ▶ Step 2C: Identify and apply patterns for each delta requirement in the AOD
 - Step 2C1: Identify delta requirements that need to be addressed in the AOD
 - Step 2C2: Identify and apply Business and Integration patterns for the delta requirements
 - Step 2C3: Identify and apply Application patterns for delta Business and Integration patterns
 - Step 2C4: Capture decisions for chosen patterns into the Architecture Decisions document
- ▶ Step 2D: Transition the AOD from Subsystem view to Logical Node view
 - Step 2D1: Seed the AOD (Logical Nodes) with logical system architecture from large reusable asset
 - Step 2D2: Identify and apply Runtime patterns for the delta Application patterns
 - Step 2D3: Integrate the Runtime pattern for the Application patterns into the AOD (Logical Nodes)
 - Step 2D4: Capture decisions for chosen patterns into the Architecture Decisions document
- ▶ Step 2E: Review and apply logical design guidelines to the AOD (Logical Nodes)
- ▶ Step 2F: Conduct technical walkthroughs to ensure that functional requirements are met

We will explain each of these steps in this section.

Step 2A: Derive the initial Architecture Overview Diagram

An Architecture Overview Diagram (AOD) is a graphical representation of the business description. It illustrates the basic ideas of the proposed architecture. Keep in mind that the objective is to keep the diagram simple, yet informative. Typically, the diagram will evolve with a greater level of detail as the architecture is better understood. The diagram serves as means of confirming an architectural understanding between the IT Architect and the client. It also serves as a basis for more detailed work products such as the Component Model and the Operational Model, as we will see later on.

The development of the AOD begins with depicting the functional blocks (Subsystems) of the solution. Functional blocks represent a logical grouping of use cases. Table 3-5 on page 99 displays the functional blocks that were

identified from the case study's use cases, which we identified earlier in this chapter. Keep this table available for reference as you continue through the process. It is important to ensure that you know what use cases are rolled into a functional block. As you continue forward, you will be applying patterns to the functional subsystem blocks (defined in the first column).

Table 3-5 Functional blocks that were identified after grouping similar use cases

Functional Subsystems	Associated Use Cases
User Management	<ul style="list-style-type: none"> ▶ Register with Agency (SS) ▶ User Provisioning (SS) ▶ Authenticate and Authorize User (Accl)
Entry Management	Request Entry (SS)
Risk Management	<ul style="list-style-type: none"> ▶ Assess Risk (EE) ▶ Generate Alert (Apia)
Inspection	<ul style="list-style-type: none"> ▶ Inspect Cargo (SS) ▶ Generate Incident (Appl)
Cross-Agency Collaboration	<ul style="list-style-type: none"> ▶ Establish Team and Environment (C) ▶ Manage and Track Incident (C) ▶ Resolve Incident (C)
Cross-Agency Information	<ul style="list-style-type: none"> ▶ Find SMEs (IA) ▶ Find Agency Information (IA)
Cross-Agency Document Management	<ul style="list-style-type: none"> ▶ Author and Approve Bulletin (C) ▶ Publish Bulletins (C)

The *User Management* functional block was created by combining the *Register With Agency*, *User Provisioning*, and *Authenticate and Authorize User* use cases. Key points to keep in mind when laying out the AOD are:

1. Identify business functions required, split into front-end (user-facing), middle (logistical), and back-end (core systems)
2. Make central business functions central to the picture
3. Leave gaps for connections and pattern boundaries
4. Put shared business functions near the middle of the picture
5. Use a dotted rectangle to aggregate related business functions together in a visual aggregation
6. Expect to iterate—This is only the beginning. The diagram will be modified as the patterns are applied

Figure 3-9 on page 100 is the AOD (Subsystem) that was developed for the case study depicting the functional blocks that were identified above.

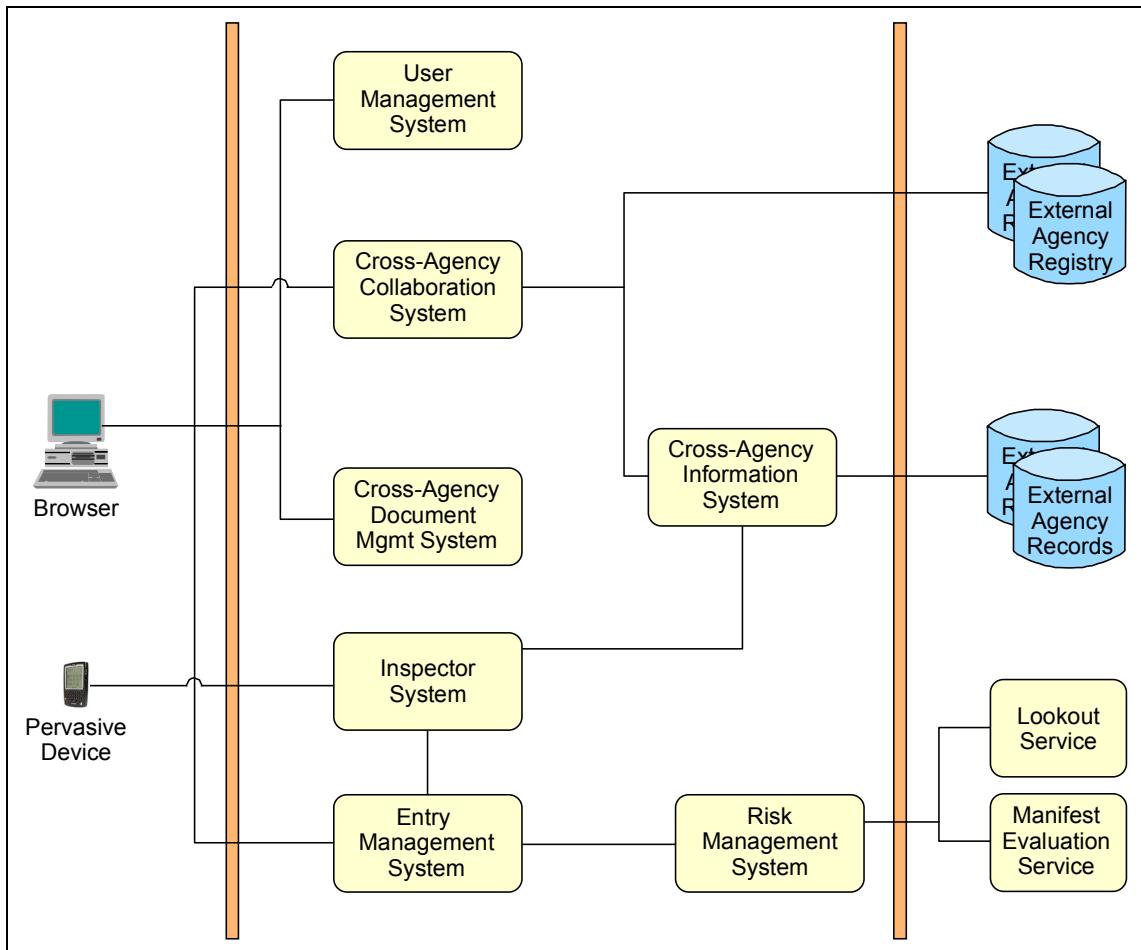


Figure 3-9 The AOD (Subsystem) begins with a display of the major business functions

Figure 3-9 depicts a preliminary AOD (Subsystem) with the major business functions identified and connected to illustrate their interactions. This will be updated as we apply the patterns in the next step. This is a good opportunity to review with the customer, confirming that you have identified all of the major business functions of the proposed system.

Note: The output of this task is the *Architecture Overview Diagram* work product.

Step 2B: Seed the AOD with large reusable assets³⁵

Figure 3-10 shows the case study's AOD (Subsystem) after the Portal composite pattern has been applied.

Note: The dashed lines represent pattern boxes that are subsumed by the Composite pattern.

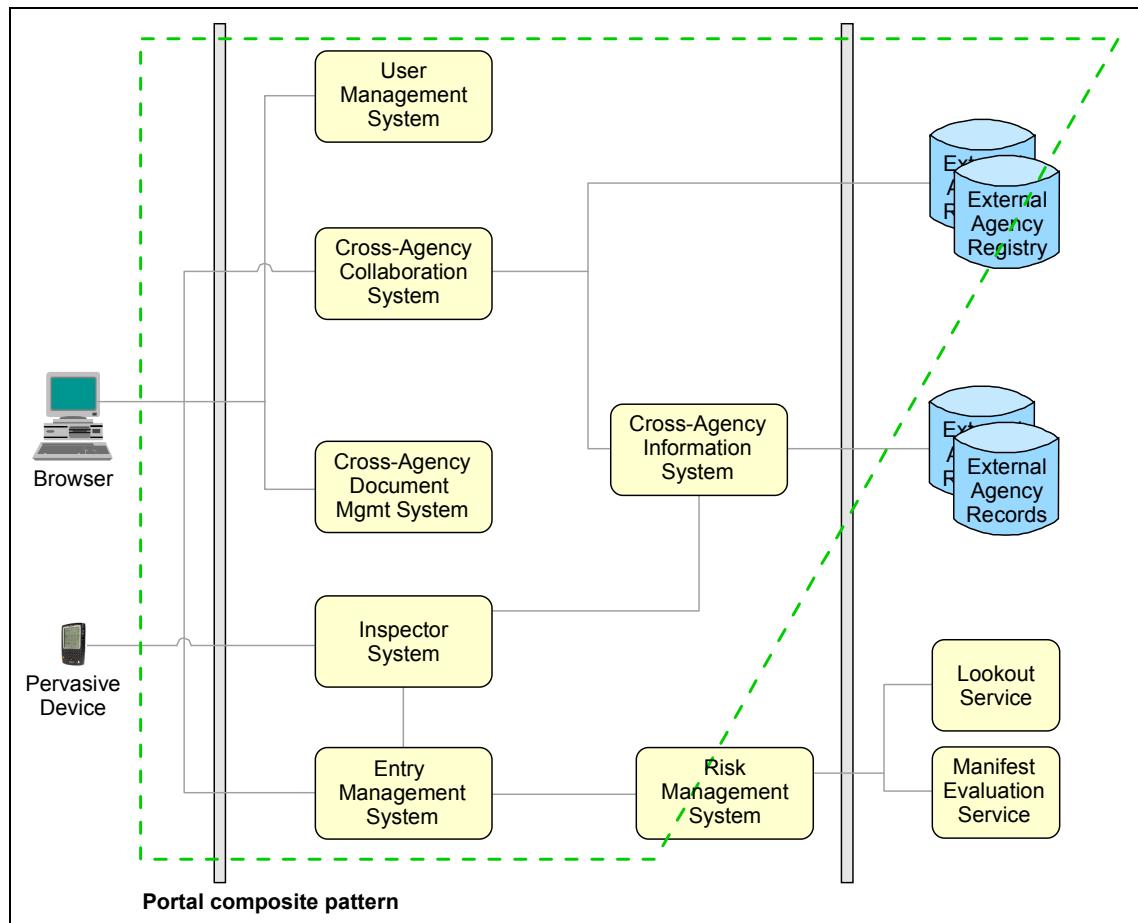


Figure 3-10 Architecture Overview Diagram (subsystem) after the Composite pattern is applied

³⁵ For example, Composite patterns, reference architectures, and industry solutions

Table 3-6 shows the functional subsystems and associated use cases that are handled by the Portal composite pattern.

Table 3-6 Subsystems/Use Cases handled by the Composite pattern

Functional Subsystems	Associated Use Cases
User Management	<ul style="list-style-type: none">▶ Register with Agency (SS)▶ User Provisioning (SS)▶ Authenticate and Authorize User (Accl)
Entry Management	Request Entry (SS)
Risk Management	<ul style="list-style-type: none">▶ Assess Risk (EE)▶ Generate Alert (Appl)
Inspection	<ul style="list-style-type: none">▶ Inspect Cargo (SS)▶ Generate Incident (Appl)
Cross-Agency Collaboration	<ul style="list-style-type: none">▶ Establish Team and Environment (C)▶ Manage and Track Incident (C)▶ Resolve Incident (C)
Cross-Agency Information	<ul style="list-style-type: none">▶ Find SMEs (IA)▶ Find Agency Information (IA)
Cross-Agency Document Management	<ul style="list-style-type: none">▶ Author and Approve Bulletin (C)▶ Publish Bulletins (C)



Sidebar: The process used to conclude Portal composite pattern applicability

The following section provides additional details on how the individual Business and Integration patterns were identified for the case study. This section is for those readers who might not be comfortable with the apparent leap to the AOD (Subsystem) shown in Figure 3-10 on page 101. Others should skip ahead to the next step in our solution development process (Step 2B1: Validate the AOD from matched large reusable asset matches requirements).

Business patterns

Business patterns are used to describe the key business purpose of a solution. These patterns describe the objectives of the solution, the high-level participants that interact in the solution, and the nature of the interactions between those participants. Structurally, these patterns are made up of at least two of the following three entities that occur in e-business solutions:

- ▶ ***Users of the solution***—This can include customers, investors, partners, vendors, and so on.
- ▶ ***Enterprise or organization with which the users interact***—This “business” entity can be used to represent the organization itself, or systems (applications or software programs) that exist within the organization
- ▶ ***Data that exists within the organization***—Data is distinguished from applications because the nature of interactions between these entities is very different.

As mentioned in [Chapter 2, “Portal solutions” on page 39](#), there are business and IT drivers that influence the selection of a pattern for a given application. You should review these drivers along with the general guidelines to help you assess if patterns are applicable. Let's go through this process now.

Identify the Business pattern

Throughout the process of applying patterns, you will use the AOD (Subsystem). Remember to keep the functional block's use cases in mind as you review each Business pattern. Let's go through an example using the Self-Service business pattern.

From the Patterns for e-business Web site, review the Self-Service business pattern. Review the pattern overview and then follow the “Guidelines” link to:

- ▶ Analyze the business and IT drivers
- ▶ Understand the general context
- ▶ Identify the major components of the pattern
- ▶ Review real-life situations where the pattern can be observed

Note: The IBM Patterns for e-business Web site can be found at
<http://www.ibm.com/developerworks/patterns/>

Self-Service business pattern

The Self-Service business pattern (also known as the *User-to-Business* or *U2B* pattern) captures the essence of direct interactions between interested parties and a business. Interested parties include customers, business partners, stakeholders, employees, and all other individuals with whom the business intends to interact. For simplicity, these interested parties are referred to as *users*. In this definition, the term *business* represents various types of organizations, including large enterprises, small and medium businesses, and government agencies.

The following are general characteristics that this pattern satisfies (in other words, the business and IT drivers):

- ▶ End-users and customers must directly interact with business processes.
- ▶ The business process must be integrated with existing business systems and information.
- ▶ The business process must be reachable in a common, consistent, and simplified manner through multiple delivery channels.

Now let's look at the case study and identify a Business pattern.

Look at the *Entry Management System* functional block on the AOD (Subsystem). Recall which use case was encapsulated into this block. Reference Table 3-7 on page 105 to keep the use case in mind as you identify the patterns.

Table 3-7 Excerpt of the Functional block table³⁶

Functional Subsystems	Associated Use Cases
Entry Management	Request Entry (SS)

Begin with Self-Service business pattern that we reviewed earlier. Keep in mind the use cases that the functional block satisfies. In this example, does the *Request Entry* use case share some of the characteristics of the Self-Service business pattern? For this functional block, Self-Service is the appropriate pattern.

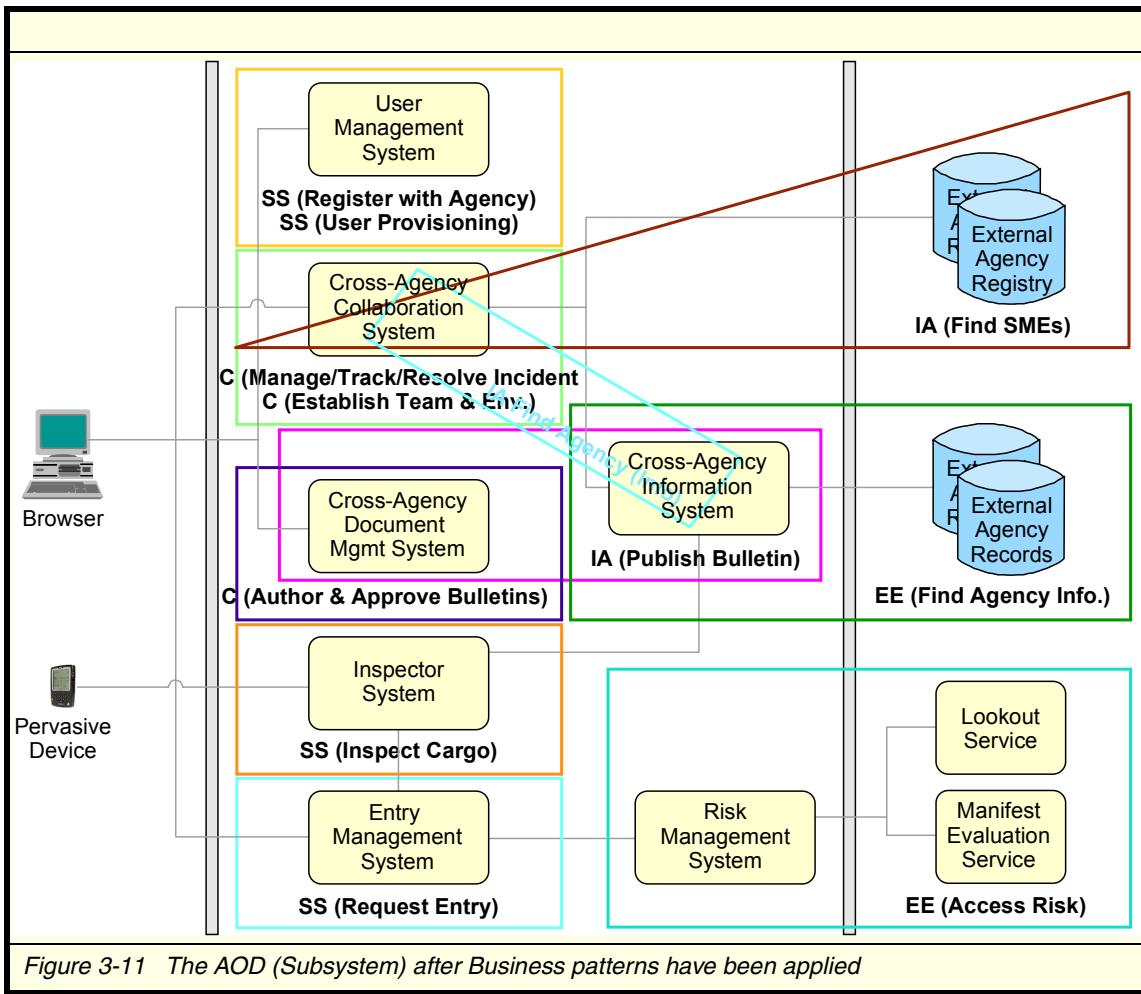
If the functional block does not fit the characteristics of this Business pattern, move on to another functional block, making note to compare the functional block to a different pattern later.

Apply the Business pattern

As you identify an appropriate Business pattern, draw pattern boxes on the AOD (Subsystem) to depict the Business pattern within the solution. Continue reviewing each functional block of the AOD (Subsystem) in order to identify which Business pattern is appropriate for each functional block until all of the blocks have been categorized.

Figure 3-11 on page 106 is the complete e-government AOD (Subsystem) after all Business patterns have been applied.

³⁶ (focusing on the “Entry Management” functional block)



In Figure 3-11 each pattern box uses the notation *BusinessPatternAbbr (Use Case Name)*—for example, **SS (Request Entry)**. This notation ties together the use case, a functional subsystem, and the pattern used to provide that functional behavior. Most of Figure 3-11 shows Business patterns that exist independently of one another and do not interact or interface with one another. However, **IA (Find Agency Info.)** is an example of a pattern that does interact with other Business patterns. The next task, applying Integration patterns, will help you better understand this example.

Identify and apply Integration patterns

Integration patterns allow for more complex solutions to be built by combining multiple Business patterns. These patterns allow the integration of multiple applications, modes of access, and sources of information to build one seamless application. Integration patterns are differentiated from Business patterns in that they do not themselves automate specific business problems. Rather, they are used within Business patterns to support more advanced functions, or to make Composite patterns feasible by allowing the integration of two or more Business patterns. (Composite patterns are discussed further in the next task.)

Whenever more than one Business pattern is needed to solve a business problem, an Integration pattern must be included so that the solution can be simplified or made seamless to the user or application requiring the solution. Therefore, Integration patterns assist in implementing the full solution by integrating Business patterns to satisfy the full e-business requirements.

Reliable integration of applications—whether legacy “stovepipe” applications, packaged software applications, or custom applications—requires the use of proven replicable patterns. At its highest level, application integration can be divided into two essentially different approaches:

- ▶ ***Process integration***—The integration of the functional flow of processing between applications.
- ▶ ***Data integration***—The integration of the information used by applications.

Neither approach is necessarily better than the other. Rather, specific integration requirements dictate which approach best solves a given business problem. For example, the integration of an e-commerce application with an Enterprise Resource Planning (ERP) system for a newly created sales order would most definitely be a process integration activity. However, in the same solution, the master data synchronization of the product catalog between the ERP system and the e-commerce system would be a data integration activity.

Business and IT Drivers

Businesses developing a solution needing the following characteristics should consider using the Application Integration business pattern:

- ▶ The business process must be integrated with existing business systems and information.
- ▶ The business processes should integrate with processes and information that exist at partner organizations.
- ▶ The business activity should aggregate, organize, and present information from various sources both within and outside of the organization.

Reference the AOD (Subsystem) again. First, look at each of the lines between the functional blocks and determine how the integration should be accomplished. You should understand the relationships between the Business patterns that have been identified within the solution. Functions that should be tied together to provide a consistent experience to the user indicate the occurrence of an Integration pattern.

Let's look at the case study and identify an Integration pattern. For the next example, we'll focus on the integration point (or connection) between the *Entry Management* and *Risk Management* functional blocks.

Remember from the use cases that the operator requests permission to dock at the port. Once that request is received, the system must assess the risk of the cargo/crew to determine if an inspection is needed.

Table 3-8 Excerpt of the Functional block table³⁷

Functional Subsystems	Associated Use Cases
Entry Management	Request Entry (SS)
Risk Management	<ul style="list-style-type: none">▶ Assess Risk (EE)▶ Generate Alert (Appl)

³⁷ (focusing on the “Entry and Risk Management” functional blocks)

Table 3-8 shows the use cases that are represented on the AOD (Subsystem) by the *Entry Management System* and *Risk Management* functional blocks. This is an example of business process integration, which helps to identify that it is an Application Integration pattern.

Next, you can extend the AOD (Subsystem) by drawing ellipses to indicate applicable Integration patterns as they are identified. Figure 3-12 shows the case study's AOD (Subsystem) after all the Business and Integration patterns have been applied. This diagram will be refined later by further identifying appropriate Application patterns.

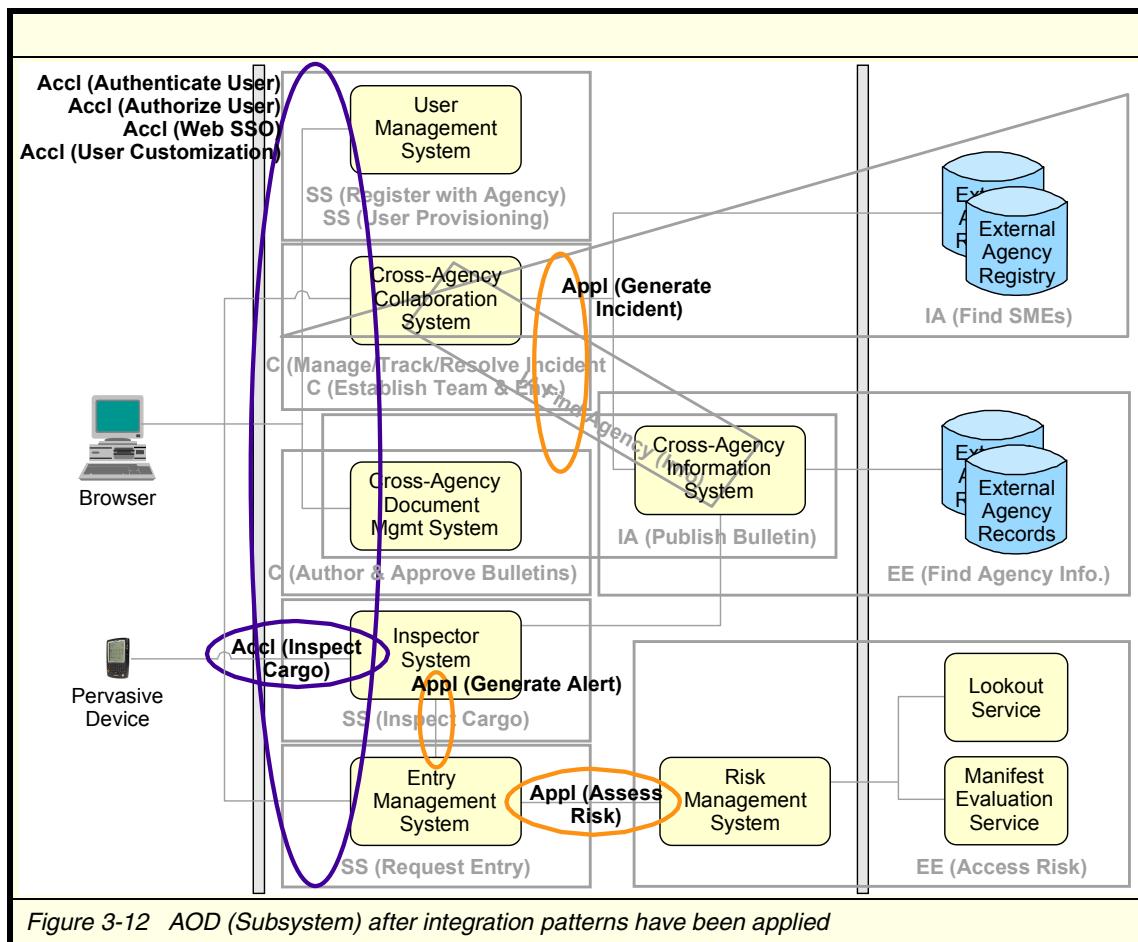


Figure 3-12 AOD (Subsystem) after integration patterns have been applied

Use the Composite pattern to identify Application patterns

The following steps outline how to use the mandatory Application patterns from a Composite pattern to identify both the Application patterns for your solution and the functional blocks that are covered by the application of the corresponding Composite pattern:

1. Review each pattern box on the AOD (Subsystem).
 - a. Focus on the use cases that are satisfied by this functional block.
2. If you have identified a pattern box (around the functional block) as a Business pattern, review that Composite pattern's mandatory Application pattern(s).

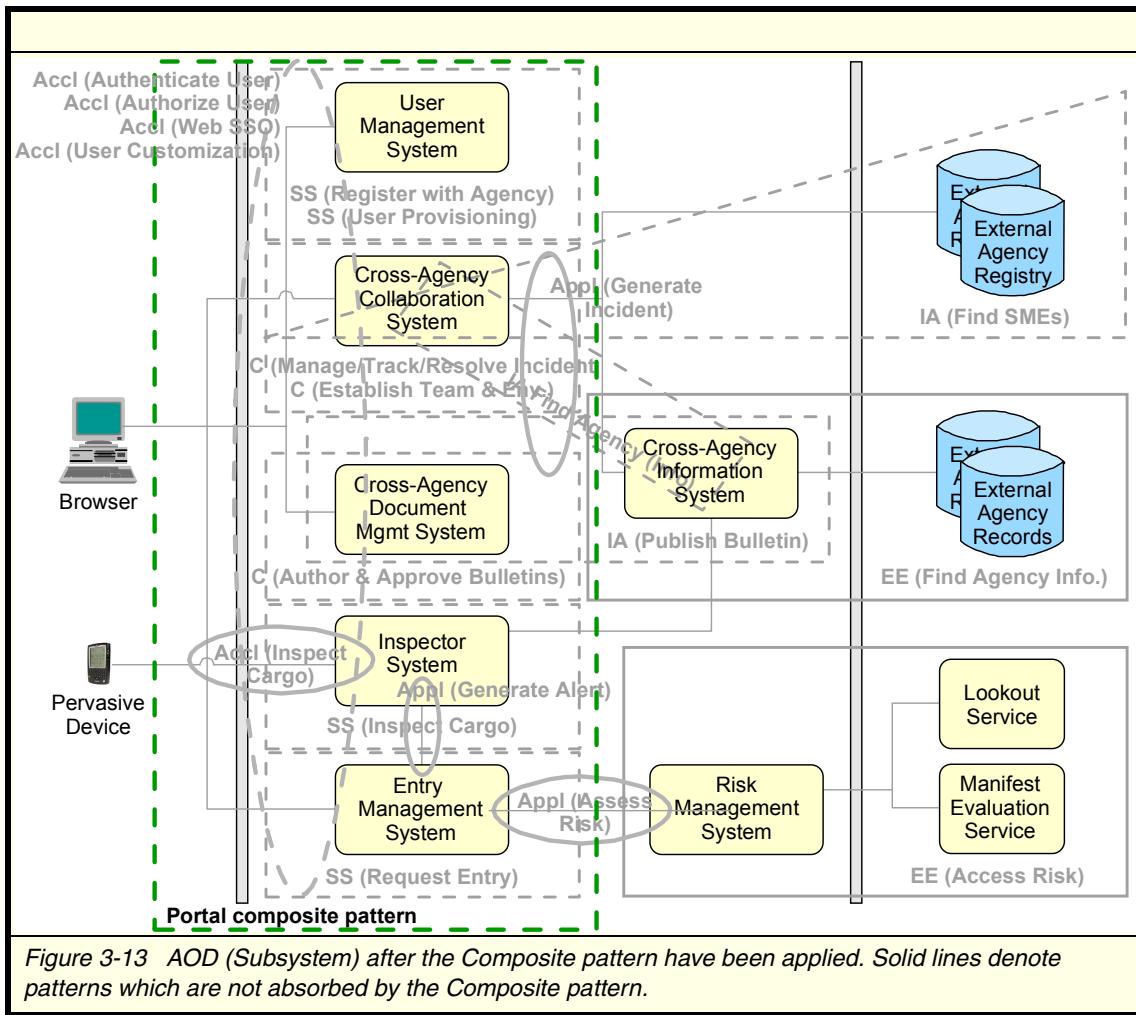
In the case study, the *Entry Management System* functional block was identified as a Self-Service business pattern. The Portal composite pattern indicates that *Directly Integrated Single Channel* is a mandatory Business pattern::Application pattern.

- a. For each mandatory Application pattern, assess the:
 - i. Business and IT drivers
 - ii. Guidelines for use
 - iii. Benefits and limitations
 - iv. Examples of where this Application pattern exists

Note: The items listed above (i. through iv.) are all available on the IBM Patterns for e-business Web site.

- b. Does this Application pattern apply to this functional block? Does it address the use cases?

Indicate on the AOD (Subsystem) which functional blocks are absorbed by the mandatory Application patterns by checking off the pattern block. For our case study, we have now generated the AOD (subsystem) shown in Figure 3-13 on page 111.



Step 2B1: Validate the AOD from matched large reusable asset matches requirements

Review the AOD (Subsystem) after the Composite pattern has been applied. Because they require one of the mandatory Business/Integration pattern::Application patterns, all of the functional blocks that have been checked off are absorbed by the Composite pattern.

In the case study, most of the AOD (Subsystem) is subsumed by the Portal composite pattern. Only some of the Access Integration patterns, Application

Integration patterns, and the Extended Enterprise business patterns fall outside the Portal composite pattern's scope.

To further validate that you have chosen the correct Composite pattern, confirm that the Application patterns associated with the selected Composite patterns are valid to your solution. To validate, review the set of mandatory and optional Application patterns associated with the Composite pattern to ensure they apply. Figure 3-14 shows the AOD (Subsystem) from the case study after all of the Application patterns have been validated.

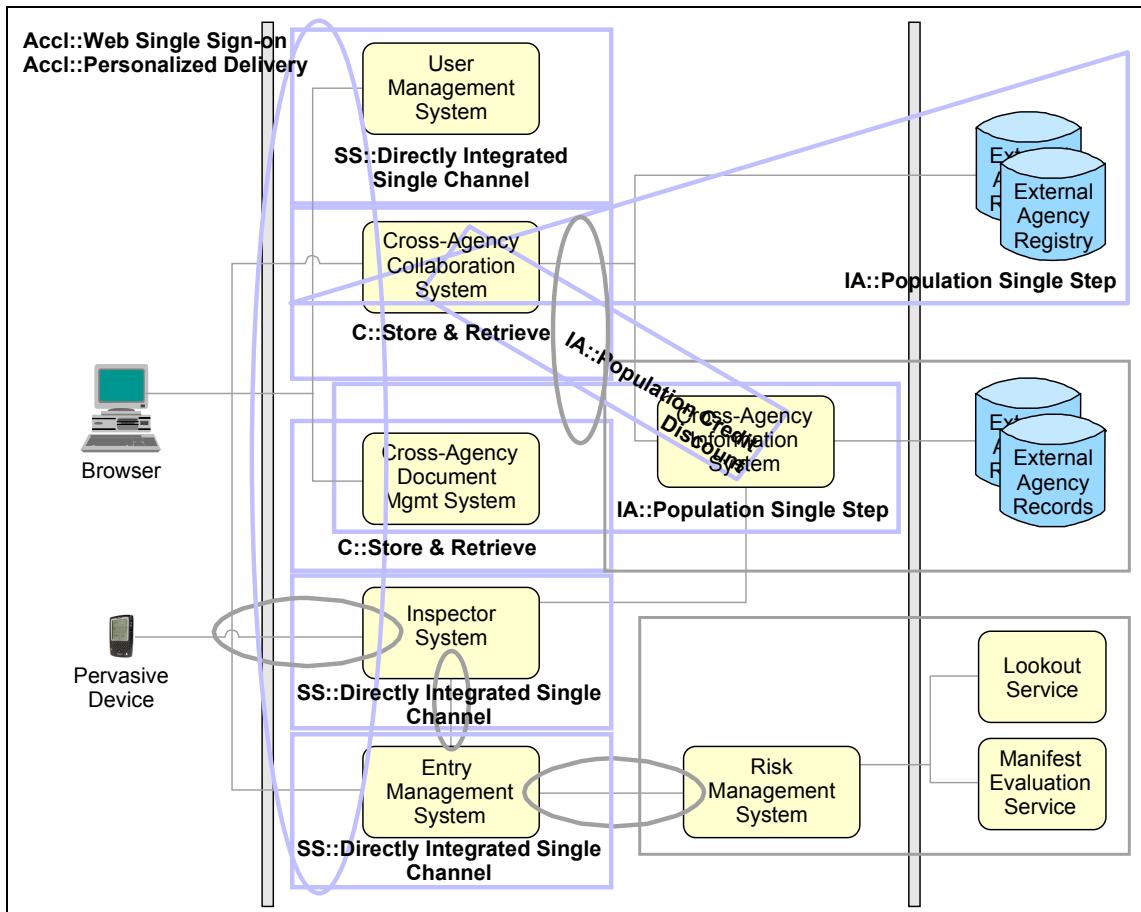


Figure 3-14 The AOD (Subsystem) after the chosen Composite pattern's Application patterns have been validated

Table 3-9 Validation of Application Patterns covered by Composite pattern for e-government case study

Application Pattern	Discussion
Accl::Web Single Sign-on	<i>Authenticate and Authorize User</i> requires a common portal-style SSO environment.
Accl::Personalized Delivery	Requirement for role-based presentation (not directly linked to a particular use case)
SS::Directly Integrated Single Channel	<p>Single delivery channel for all SS applications (the presentation server). UI to be adapted for environment (in other words, not <i>As-is Host</i>).</p> <p><i>Included use cases:</i></p> <ul style="list-style-type: none"> ▶ Register with Agency ▶ Request Entry ▶ User Provisioning ▶ Inspect Cargo
C::Store and Retrieve	<p>This covers e-mail, SMS messaging, simple chat via IRC, and document-sharing via TeamRooms.</p> <p><i>Included use cases:</i></p> <ul style="list-style-type: none"> ▶ Manage and Track Incident ▶ Resolve Incident ▶ Author and Approve Bulletins
IA::Population Crawl and Discovery	Many documents (unstructured data) needed to be searched to populate TeamRoom. No summarization needed. Included use case is <i>Find Agency Info</i> .
IA::Population Single Step	No data needs to be transformed or restructured. Included use cases are <i>Find SMEs</i> (structured data) and <i>Publish Bulletin</i> (unstructured data)

It is important that you go through these steps to assess if your solution truly resembles the Composite pattern you assumed. Looking at Figure 3-15 on page 114, you might think that the Portal composite pattern could be applied. It has the necessary Self-Service, Collaboration, and Information Aggregation business patterns, as well as the required Access Integration pattern, and it has some of the optional patterns as well. Simply looking at this AOD (Subsystem), you might think it's a variation of the Portal composite pattern.

However, the following illustrates the importance of paying close attention to the use cases as you go through this process. If you examine the use cases in Table 3-10 on page 115 that go along with the following solution, you will clearly see it is *not* a Portal composite pattern, even though the AOD (Subsystem) diagram in Figure 3-15 on page 114 might lead you to that conclusion. Rather, this is an example of the Electronic Commerce composite pattern.

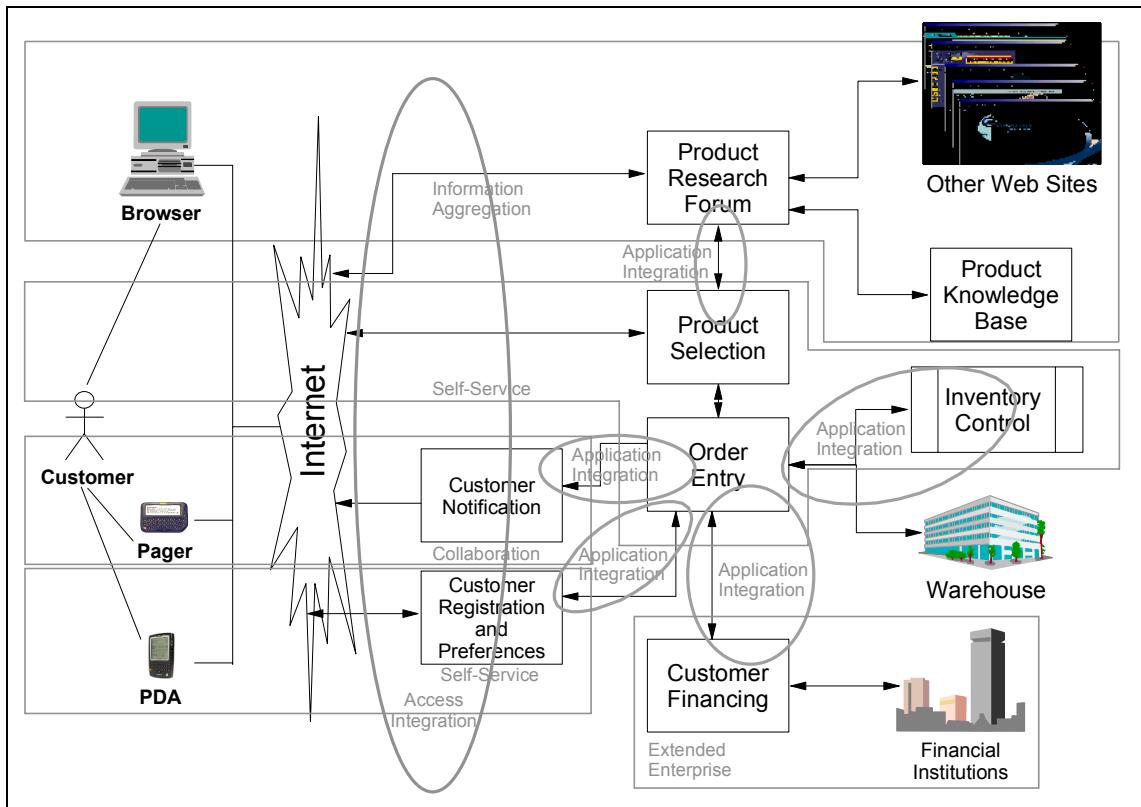


Figure 3-15 Example of an AOD (Subsystem) that could be deceiving, leading you to select a Portal composite pattern instead of the Electronic Commerce composite pattern

Table 3-10 Use cases for commerce case study

Use Case	Description
Register with Company	Customers register their names, locations, shipping addresses, preferred financial institutions, and other relevant information with the system
Perform Account Maintenance Functions (user provisioning)	Maintaining account information (addresses, etc.), managing passwords, and setting user interface parameters
Browse and Select from Catalog	Allow users to search and select components from the product catalog
Access Forum Information	Allows users to learn more about the experiences of other manufacturers, structured content providing the key design specifications, assembly advice, and reliability and desirable service strategies appropriate for their line of finished products for particular product lines
Place Order	Users will be able to place orders from the product catalog
Query Inventory	The system will verify that the component is available by querying the in-house inventory control application
Choose Financing Option	Select from several financing options provided by different financial institutions
Access Customer Preferences	Access the customer's preferences and other profile information from the customer registration database to complete the details of the order
Notify Customer	Once the order is placed, customers will be notified electronically via e-mail or pager
Send Order to Warehouse	The completed order is then forwarded via e-mail to the warehouse to be filled and shipped to the customer

Step 2B2: Capture decisions for chosen patterns into the Architecture Decisions document

We have reminded you at various points in this process of the need to document the architectural decisions you have made. As a formal process task, we have called it out to ensure that the Architecture Decisions document is updated.

Here is an example of the Architecture Decisions document for the case study. Table 3-11 on page 116 is an excerpt of the Architecture Decisions document that focuses on the decisions made when choosing the Information Aggregation for the *Find SMEs* functional block of the case study.

Table 3-11 Excerpt of the Architecture Decisions document

Subject Area	Expertise Operational Datastore
Architectural Decision	Access to a logical (read-only) operational data store is required to support <i>Find SMEs</i> .
Issue or Problem	Provide information aggregation to quickly and efficiently locate SMEs to participate on the X-Agency Response Team.
Assumptions	Requirement for <i>Find SMEs</i> is a read-only access to LDAP information.
Motivation	Minimize network traffic and latency when setting up a new incident team.
Alternatives	Could make multiple queries (crawl and discover) at team creation time.
Decision	Choose Population Single Step (application pattern) to generate a composite "inter-agency" SME ODS.
Justification	Low volatility in ODS records. Only require read-only access to data.
Implications	
Derived Requirements	
Related Decisions	

This document will continue to be updated as you make other decisions throughout this process. The format shown in Table 3-11 is an example of how to document such decisions

Step 2C: Identify and apply patterns for each delta requirement in the AOD

Consists of the following sub-steps:

- ▶ Step 2C1: Identify delta requirements that need to be addressed in the AOD
- ▶ Step 2C2: Identify and apply Business and Integration patterns for the delta requirements
- ▶ Step 2C3: Identify and apply Application patterns for delta Business and Integration patterns
- ▶ Step 2C4: Capture decisions for chosen patterns into the Architecture Decisions document

Step 2C1: Identify delta requirements that need to be addressed in the AOD

Review the AOD, this time focusing on the pattern boxes/integration points that were unabsorbed by the Portal composite pattern.

In the e-government case study's AOD (Subsystem), look at the *Risk Management System* pattern box. This is an example of a Business pattern that falls outside the Portal composite pattern because it is an Extended Enterprise business pattern. This Business pattern is an optional pattern within the Portal composite pattern. Its pattern box is thus identified as a delta requirement.

Similarly, the line between *Entry Management System* and *Inspection System* functional blocks was identified as an Application Integration pattern. The Portal composite pattern does not mandate that the Application Integration pattern be used, so that this pattern is identified as a delta requirement. Make a mental note of this as we continue through these steps.

After reviewing all of the delta requirements, we can identify the delta Business and Integration patterns (as illustrated by Figure 3-16).

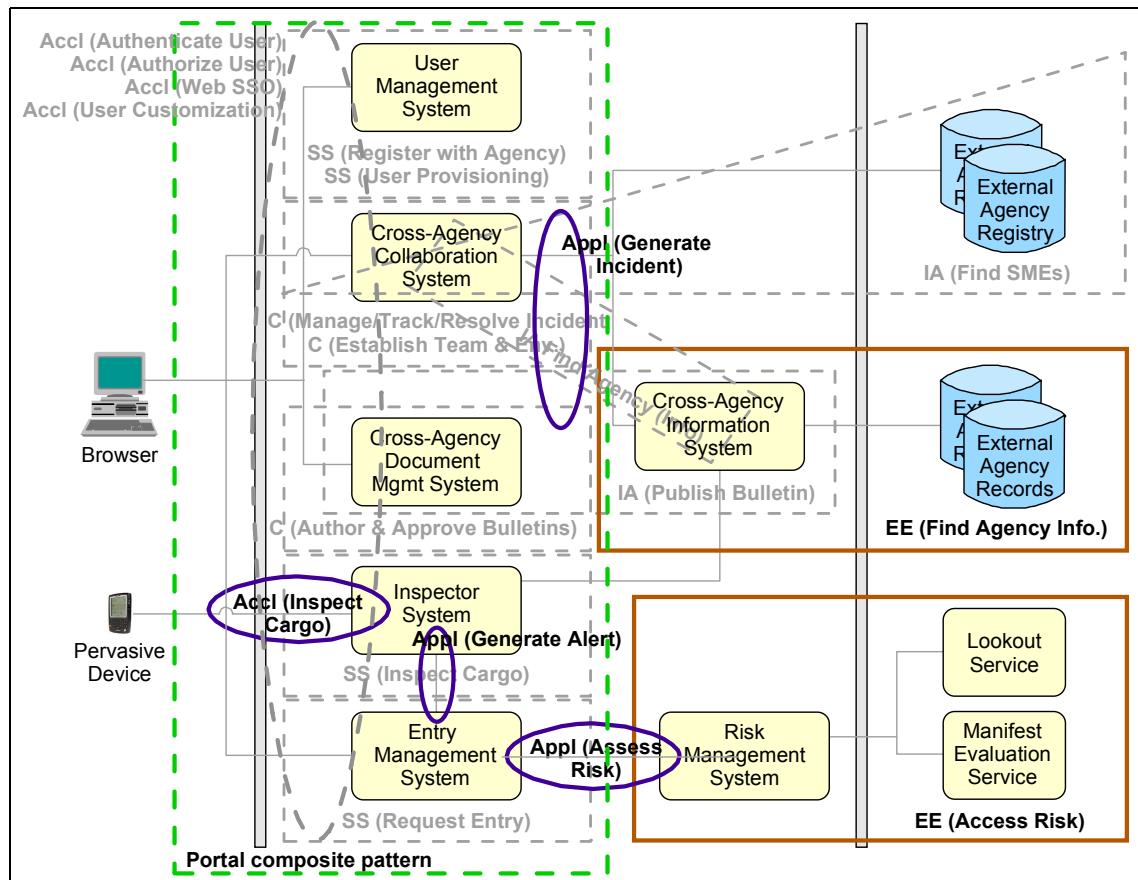


Figure 3-16 AOD (Subsystem) after delta requirements have been highlighted

Delta requirements will be our primary focus throughout the remainder of the solution design process.

Step 2C2: Identify and apply Business and Integration patterns for the delta requirements

Depending on the analysis performed thus far in Step 2, you may have already identified Business and Integration patterns associated with delta requirements. For example in the case study illustrated by Figure 3-16 on page 117, all of the deltas have been classified into the appropriate Business or Integration pattern.

This formal task has been defined to address the situation where your large reusable asset has not yet identified patterns. In order to continue to the next step of the process, it is important that you have these patterns identified. The process applied here is the standard process identified in *A Portal Composite Pattern Using WebSphere V4.1*.³⁸

Step 2C3: Identify and apply Application patterns for delta Business and Integration patterns

The next step involves identifying the Application patterns associated with each of the delta Business and Integration patterns. You should use the standard approach of reviewing the business and IT drivers associated with each potential Application pattern and comparing these to your solution requirements.

For the case study, the following are the deltas that were captured in the AOD (Subsystem):

1. Access Integration (Inspect Cargo)
2. Application Integration (Assess Risk)
3. Application Integration (Generate Alert)
4. Application Integration (Generate Incident)
5. Extended Enterprise (Assess Risk)
6. Extended Enterprise (Find Agency Information)

As an example, let's focus on the Extended Enterprise (Assess Risk) delta. The general problem addressed by this pattern is illustrated by Figure 3-17 on page 119.

³⁸ IBM Redbook SG24-6869-00 (ISBN 0738427446)

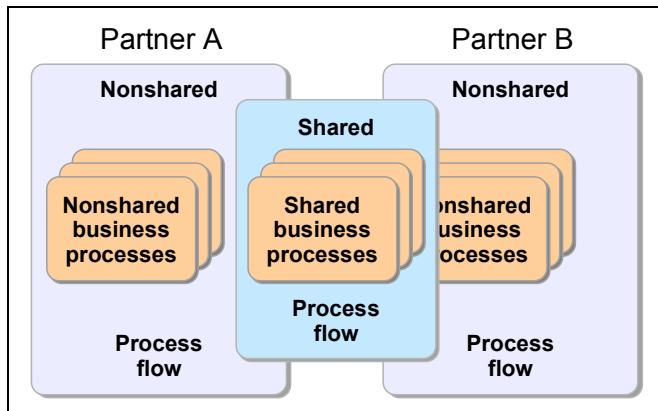


Figure 3-17 Illustration of the general problem addressed by the Extended Enterprise pattern

Interactions between partners form a public process (or potentially multiple distinct public processes). Each of these must be integrated into the private business process flows implemented by each partner. This separation of public and private processes provides the benefit of flexibility and autonomy for the trading partners.

In order to identify the appropriate Application pattern, you need to assess the business and IT drivers of the solution to the Extended Enterprise::Application patterns.

Let's recall the business and IT drivers for the case study again:

Business drivers

- ▶ Need to integrate “stovepipes” and decrease redundancy
- ▶ Need to streamline processes and improve data analysis
- ▶ Need to facilitate effective decision-making

IT drivers

- ▶ Better internal use of information
- ▶ Secure data sharing
- ▶ Effective incident management through automated processes
- ▶ Support new security demands
- ▶ Interface to new and existing databases

With these in mind, we'll review the business and IT drivers associated with the various Extended Enterprise business pattern's Application patterns, looking for the Application patterns that satisfy most of the case study's needs. First, let's review the business drivers of the Extended Enterprise in Table 3-12.

Table 3-12 The Business Drivers for the Extended Enterprise::Application patterns

Business Drivers	Document Exchange	Exposed Application	Exposed Business Services	Managed Public Processes	Managed Public and Private Processes
Improve the organizational efficiency	✓	✓	✓	✓	✓
Reduce the latency of business events	✓	✓	✓	✓	✓
Support a structured exchange with business partners	✓	✓	✓	✓	✓
Support partner real-time access to/from applications		✓	✓	✓	✓
Support partner real-time access to/from business services			✓	✓	✓
Support for shared public process flows with partners				✓	✓
Support for internal private and shared public process flows					✓

It is important to assess which of these drivers will address your needs. Table 3-13 shows how we have tied the pattern's drivers back to the case study's in order to identify which Application pattern is most appropriate.

Table 3-13 Side-by-side look at the pattern's business drivers as opposed to the case study's

Pattern's Business Drivers	Applicable Case Study's Business Drivers
Improve organizational efficiency	<ul style="list-style-type: none"> ▶ Need to integrate "stovepipes" and decrease redundancy ▶ Need to streamline processes and improve data analysis ▶ Need to facilitate effective decision-making
Reduce the latency of business events	<ul style="list-style-type: none"> ▶ Need to streamline processes and improve data analysis ▶ Need to facilitate effective decision-making
Support a structured exchange with business partners	Need to integrate "stovepipes" and decrease redundancy
Support partner real-time access to and from applications	Need to streamline processes and improve data analysis
Support partner real-time access to and from business services	Need to streamline processes and improve data analysis
Support shared public process flows with partners	None apply
Support internal private and shared public process flows	None apply

Table 3-14 on page 122 highlights the applicable business and IT drivers associated with *Assess Risk*. Note that neither *Document Exchange* nor *Exposed Application* can satisfy the solution requirements.

Table 3-14 Highlighting the pattern's applicable business drivers

Business Drivers	Document Exchange	Exposed Application	Exposed Business Services	Managed Public Processes	Managed Public and Private Processes
Improve the organizational efficiency	✓	✓	✓	✓	✓
Reduce the latency of business events	✓	✓	✓	✓	✓
Support a structured exchange with business partners	✓	✓	✓	✓	✓
Support partner real-time access to/from applications		✓	✓	✓	✓
Support partner real-time access to/from business services			✓	✓	✓
Support for shared public process flows with partners				✓	✓
Support for internal private and shared public process flows					✓

Similarly, Table 3-15 on page 123 illustrates the applicable IT drivers.

Table 3-15 Highlighting the pattern's applicable IT drivers

IT Drivers	Document Exchange	Exposed Application	Exposed Business Services	Managed Public Processes	Managed Public and Private Processes
Leverage existing skills	✓	✓	✓	✓	✓
Leverage legacy investment	✓	✓	✓	✓	✓
Backend application integration		✓	✓	✓	✓
Minimize application complexity	✓	✓	✓	✓	✓
Minimize enterprise complexity			✓	✓	✓
Avoid partner-mandated infrastructures				✓	✓
Reduce partner dependency on specific applications			✓	✓	✓
Reduce partner dependency on specific business protocols				✓	✓

Looking at these two diagrams that highlight the applicable drivers, you can deduce which Application patterns satisfy the case study. Only three Application patterns address all of its requirements. Figure 3-18 on page 124 is a graphical representation of all five Application patterns, highlighting the best candidates.

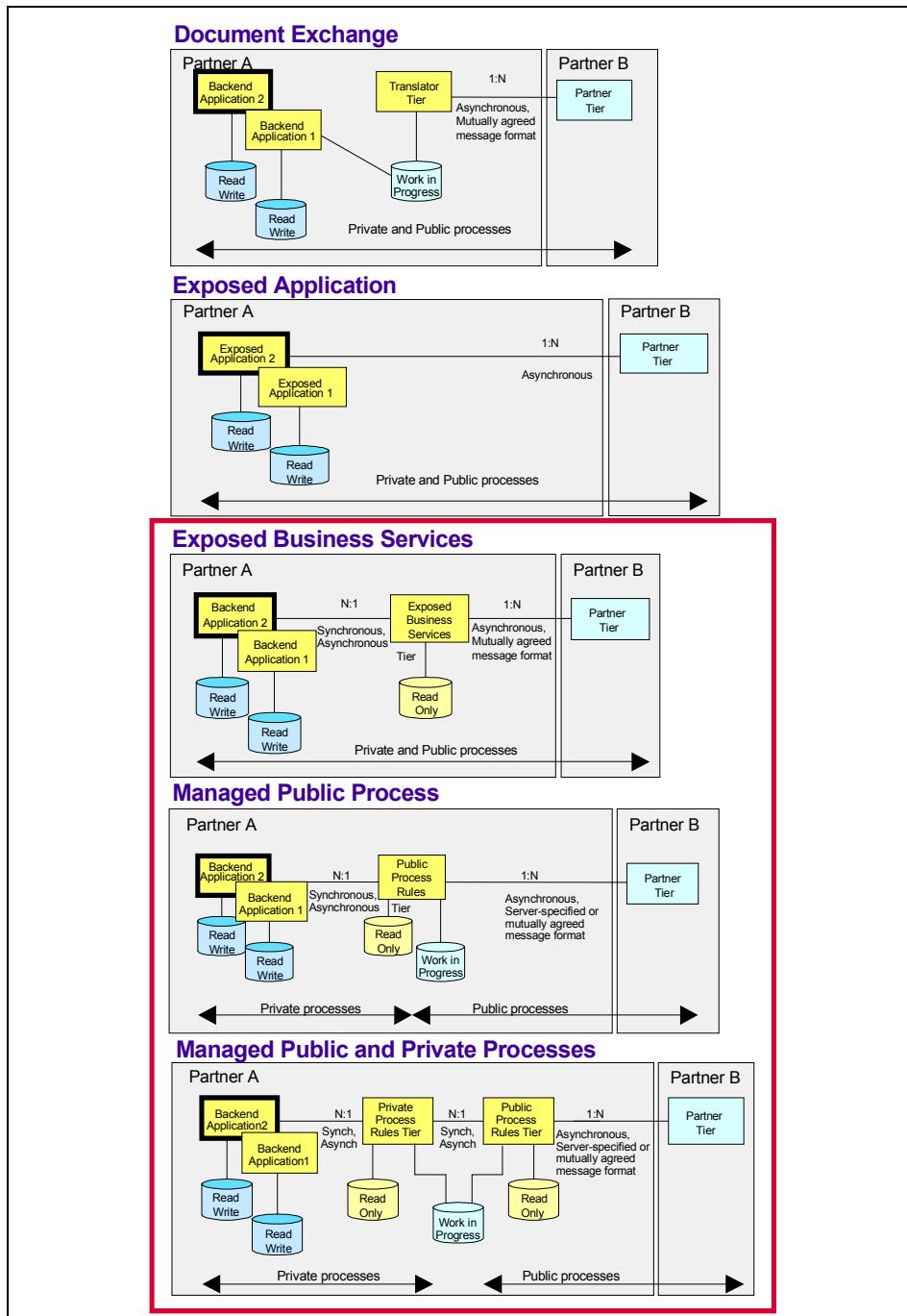


Figure 3-18 The five Extended Enterprise::Application patterns, highlighting the three that are applicable

Table 3-16 highlights more information that will help us determine the most appropriate Application pattern.

Table 3-16 Description of the various Application patterns³⁹

Application Pattern	Issues
Document Exchange	This is the ideal Application pattern to choose if your current business needs would be satisfied by the batched exchange of electronic documents. (EDI is a good example)
Exposed Application	Typically supported by asynchronous communication mechanism. Requires security features such as authentication, authorization, confidentiality, integrity and logging for non-repudiation purposes. Limitation—common middleware between all partners.
Exposed Business Services	This application pattern is ideal when there is need to hide the back-end application details from trading partners to gain flexibility and improve maintainability.
Managed Public Process	<i>Trading partner agreements.</i> The primary reason for choosing this application pattern is to support different business protocols with different trading partners. This provides greater flexibility.
Managed Public and Private Process	This Application pattern accommodates long-running transactions across organizational boundaries.

Table 3-16 on page 125 helped us drill down to the Exposed Business Services application pattern. We chose the Application pattern that is the best choice for the case study, but that does not over-complicate the solution. Because there is no public, well-defined process, the Managed Public Process and the Managed Public and Private processes would be inappropriate. Thus, the simpler of the three candidate Application patterns is the Exposed Business Services. If a well-defined trading partner agreement was later obtained, this may need to be re-evaluated.

This process should be applied for all deltas. Figure 3-19 on page 126 is the AOD of the case study showing the applied Applications patterns, while Table 3-17 on page 126 summarizes the Application patterns selected for the delta Business and Integration patterns

³⁹ (highlighting the most appropriate one for the case study)

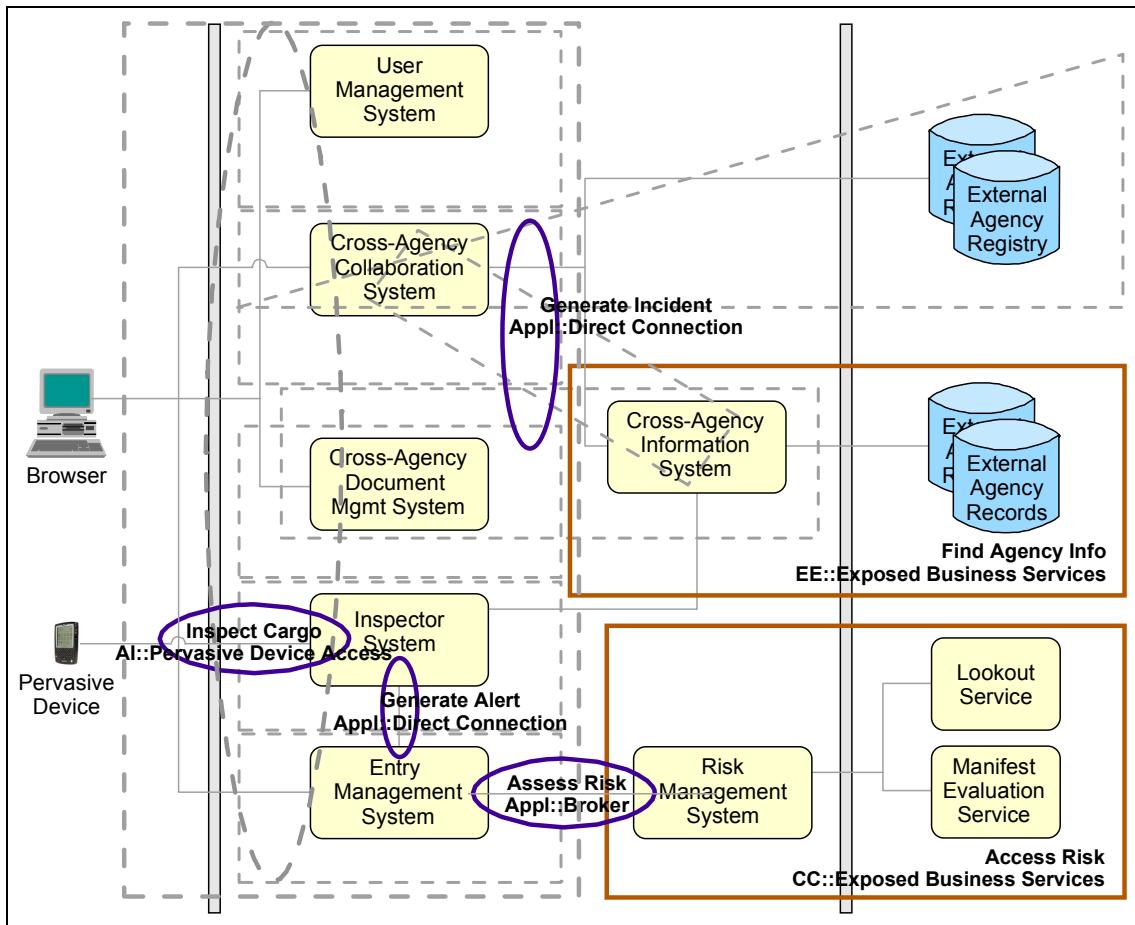


Figure 3-19 Application patterns for delta requirements

Table 3-17 Summary of Application patterns for delta Business and Integration patterns

Delta requirements in the AOD (Subsystem)	Applied Pattern
Application Integration (Assess Risk)	Application Integration::Broker
Access Integration (Inspect Cargo)	Access Integration::Pervasive Device Access
Extended Enterprise (Find Agency Information)	Extended Enterprise::Exposed Business Service
<ul style="list-style-type: none"> ► Application Integration (Generate Alert) ► Application Integration (Generate Incident) 	Application Integration::Direct Connection

Step 2C4: Capture decisions for chosen patterns into the Architecture Decisions document

Again we have created a formal process task to remind you of the importance of documenting your architectural decisions. Table 3-18 is an excerpt of the Architecture Decisions document that focuses on the decisions made when choosing the Information Aggregation for the *Find SMEs* functional block of the case study.

Table 3-18 An excerpt of the Architecture Decisions document

Subject Area	Inter-Agency Business Services
Architectural Decision	Application Integration (Assess Risk)
Issue or Problem	This is a process (rather than data) integration activity
Assumptions	The integration involves multiple systems
Motivation	<p>Application pattern choices are:</p> <ul style="list-style-type: none">▶ Direct Connection ¹▶ Transactional ²▶ Broker▶ Managed Process ³ <p>¹ Since the integration involves multiple systems, <i>Direct Connection</i> is inappropriate. ² Since there are no updates to any of the contributing applications, the integration isn't <i>Transactional</i>. ³ <i>Managed Process</i> (BPM) is typically associated with long running processes. This is short-term.</p>
Alternatives	
Decision	Applications are provided via the Broker application pattern.
Justification	
Implications	
Derived Requirements	
Related Decisions	

This document will continue to be updated as you make other decisions throughout this process. The above format is an example of how to document such decisions.

Step 2D: Transition the AOD from Subsystem view to Logical Node view

From a Patterns approach, the next step in our solution design process is to move to the Runtime pattern level. Unfortunately, the Subsystem view of our AOD is not consistent with the Logical Node view associated with Runtime patterns. This step transitions the AOD from a Subsystem view to a Logical Node view. This is a logical progression leading to physical design and the development of the Component and Operation models.

Step 2D1: Seed the AOD (Logical Nodes) with logical system architecture from large reusable asset⁴⁰

The next step is to transition the AOD to work with the Logical Node view. The large reusable asset should have a similar work product with which you can seed, building out the AOD (Logical Nodes). When working with the Portal composite pattern, the corresponding Runtime pattern can be used. Remember that if there is a portion of the matched large reusable asset that is not applicable to the current solution being developed (a delta that will not be included in the solution), the corresponding logical nodes should not be included in the seeded AOD (Logical Nodes).

Definition: *Runtime patterns* are used to define the logical middleware—in other words, to support the Application patterns. They describe the logical architecture needed to implement the Application patterns by exposing the major middleware nodes, their roles, and the interfaces among them. These logical nodes can then drive the physical model through Best Practices, Usage Guidelines, Reference Implementations, and Personal Experience.

Following Application pattern validation in the case study, we verified that the Portal composite pattern is a good starting point. Since it absorbed a major portion of our solution, it's appropriate to use the Portal composite's Runtime pattern to seed the AOD (Logical Nodes). Figure 3-20 on page 129 is the Portal composite pattern's Runtime pattern. We will use it to seed the AOD (Logical Nodes) for the case study.

⁴⁰ For example, Composite pattern's runtime pattern

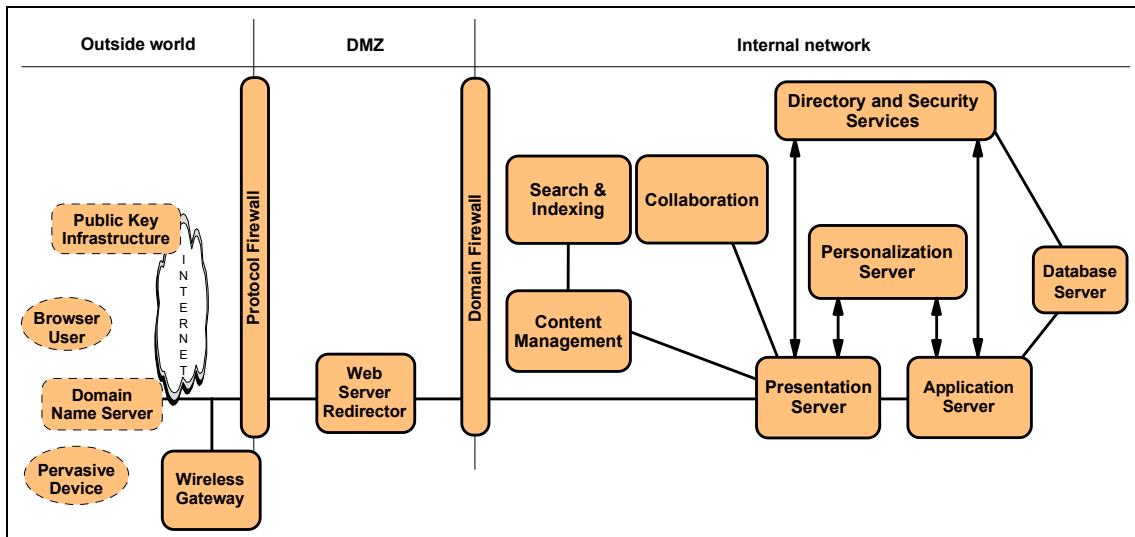


Figure 3-20 The Portal composite pattern's Runtime pattern

Note: For an explanation of each of the nodes, refer to section 5.2.1 of the redbook *A Portal Composite Pattern Using WebSphere 4.1*, SG24-6869.

The Portal composite Runtime pattern is a combination of a number of different Runtime patterns. The characteristics of the Runtime patterns and the addition of portal-specific nodes is what comprise the Portal composite Runtime pattern. During the seeding process, you may want to customize the logical nodes to better represent the domain of your solution.

Customizing the Portal composite Runtime pattern results in an initial AOD (Logical Nodes) for our solution. Figure 3-21 on page 130 represents the customized AOD (Logical Nodes) for the case study that was seeded from the reusable asset.

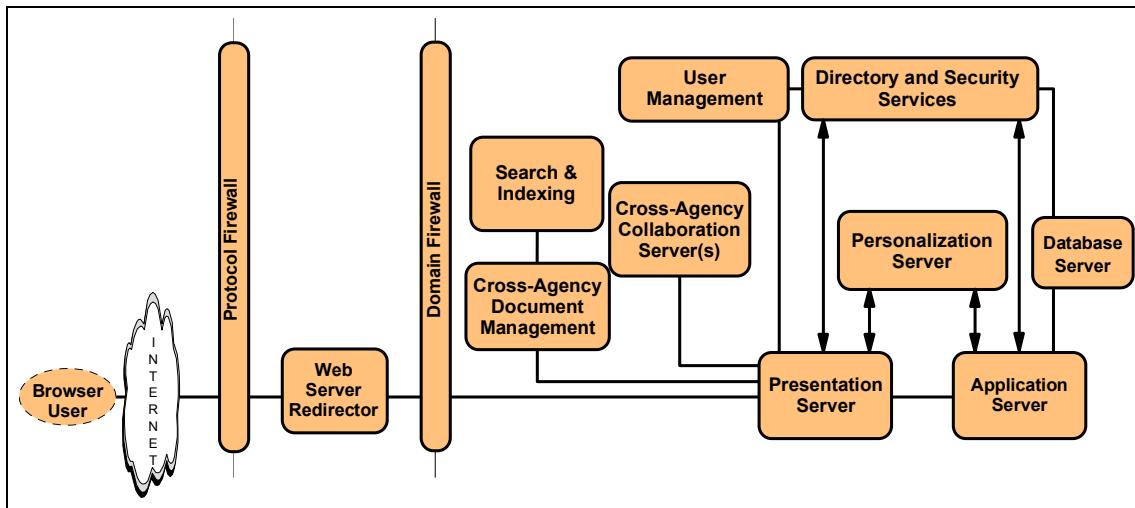


Figure 3-21 The AOD (Logical Nodes) as seeded from the customized Portal composite pattern's Runtime pattern

This customization consisted of renaming the logical nodes to be more applicable to the case study. The *User Management* node was also identified as a separate functional node in order to highlight the user registration, authentication, and authorization.

Step 2D2: Identify and apply Runtime patterns for the delta Application patterns

In order to design a complete solution, you will need to update the AOD (Logical Nodes) to include the delta Application patterns. Starting with the Application patterns for each delta requirement, you should identify the appropriate Runtime pattern to be used in the solution.

You should then review the business and IT drivers, benefits, and limitations, pulling the pattern-to-use sections from the patterns Web site in order to identify the most applicable pattern. Be sure to review all of the Runtime patterns in order to make the best choice.

Once you have chosen *Exposed Business Services* as the Application pattern used to handle *Assess Risk*, you must choose the corresponding Runtime pattern. Note there may be several “variations”—which will lead to alternative designs. You may be able to dismiss some alternatives immediately, while at others the alternatives must be “considered” until later in the design process.

Figure 3-22 shows the specific business functionality supported by the Extended Enterprise business pattern⁴¹ that we have chosen.

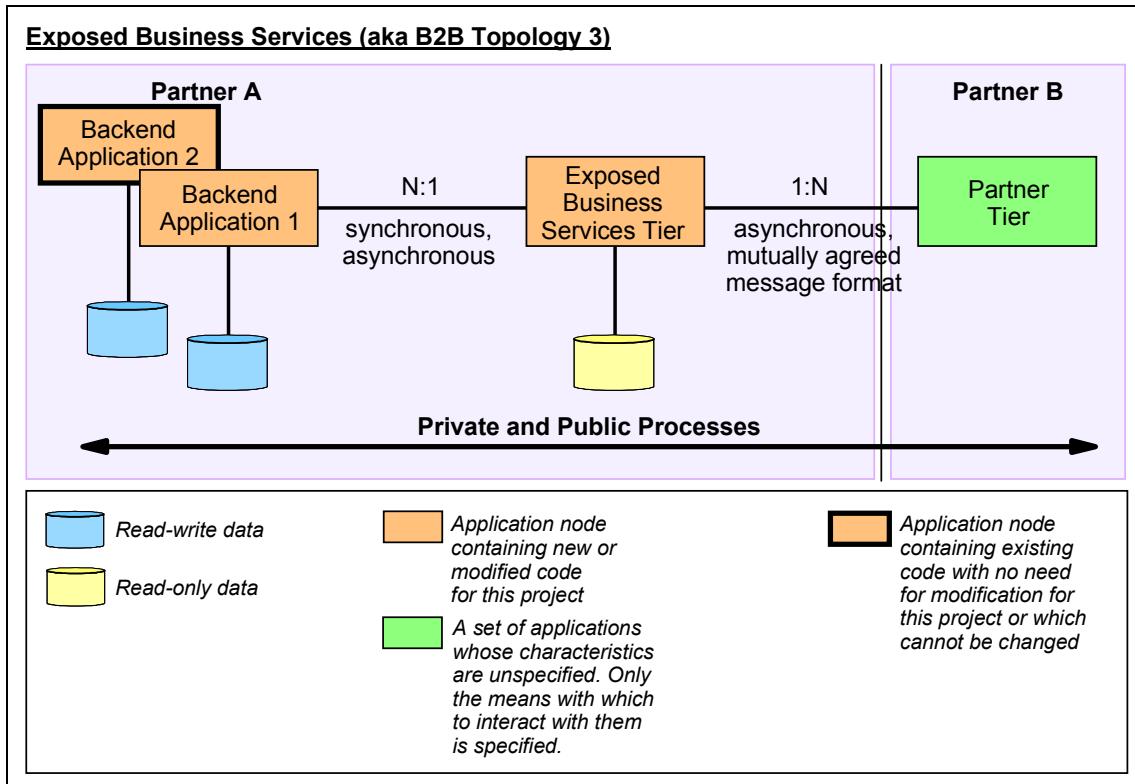


Figure 3-22 Detail of the Exposed Business Services application pattern

There are various topologies available for the Application patterns that vary from one industry to another. A survey of such applications in multiple industries, however, reveals certain common approaches that have been successful. The diagrams are used to describe these successful approaches for the Application patterns.

For Exposed Business Services, the resulting runtime pattern illustrated by Figure 3-23 on page 132 will result in adding the Message Broker node and the VPN end point to the current AOD (Logical Nodes). Figure 3-23 on page 132 is the Runtime pattern for the Exposed Business Services application pattern obtained from the Patterns for e-business Web site.

⁴¹ Also known as the Business to Business or B2B business pattern

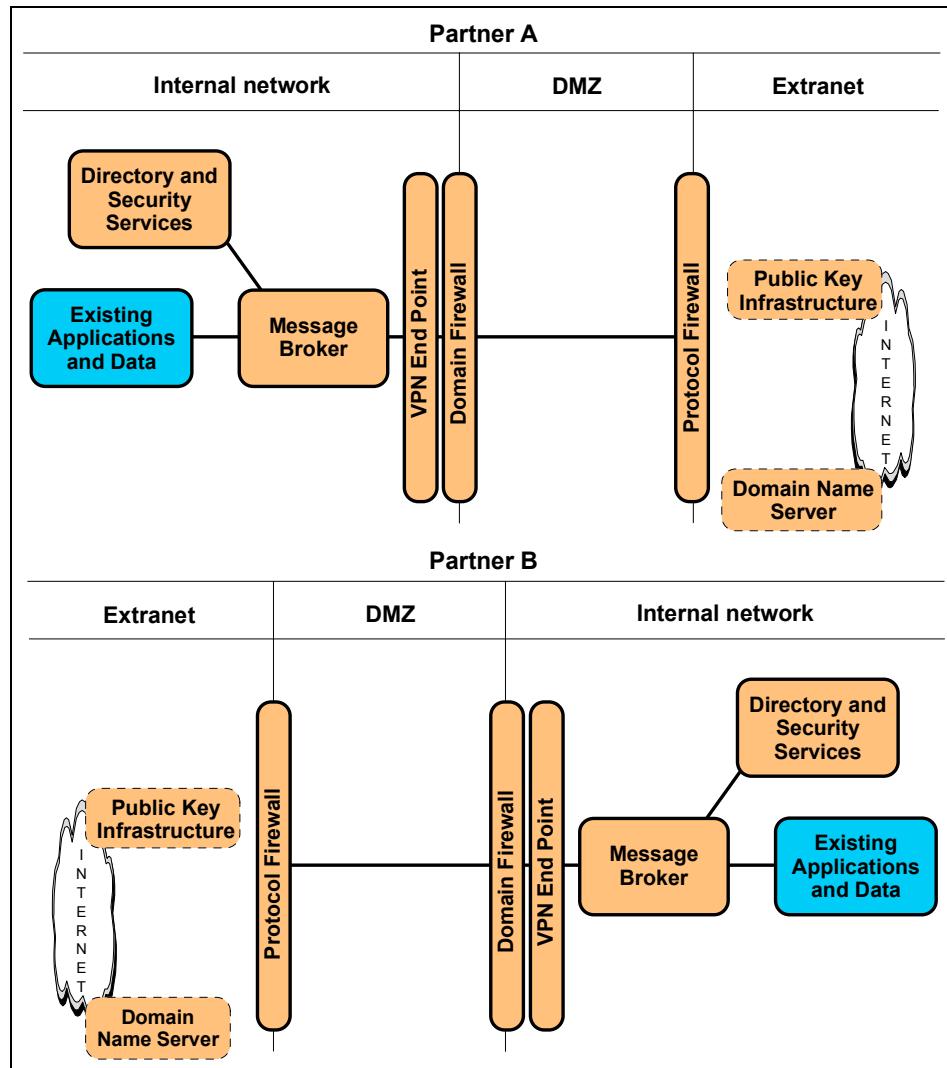


Figure 3-23 Exposed Business Services application pattern:: Runtime pattern

The Exposed Business Services Runtime pattern supports message-oriented interactions using a *Message Broker*. This broker handles enterprise application integration and communicates over a Virtual Private Network (VPN). The VPN is terminated behind the domain firewall for greater security.

Repeat this process, identifying the appropriate Runtime pattern for each delta pattern by identifying the logical nodes that are needed to support each Runtime pattern.

Step 2D3: Integrate the Runtime pattern for the Application patterns into the AOD (Logical Nodes)

Once you have identified the Application pattern and the most appropriate Runtime pattern, you will need to ensure that the solution's AOD (Logical Nodes) reflects those decisions. Review the AOD (Logical Nodes) and validate that the solution contains the logical nodes required by the chosen Runtime pattern.

The original AOD (Logical Nodes) that we created for the case study did not have the necessary logical nodes required by this Runtime pattern. Figure 3-24 illustrates the updates made to the AOD (Logical Nodes) once the Extended Enterprise Application pattern has been applied for the *Assess Risk* function.

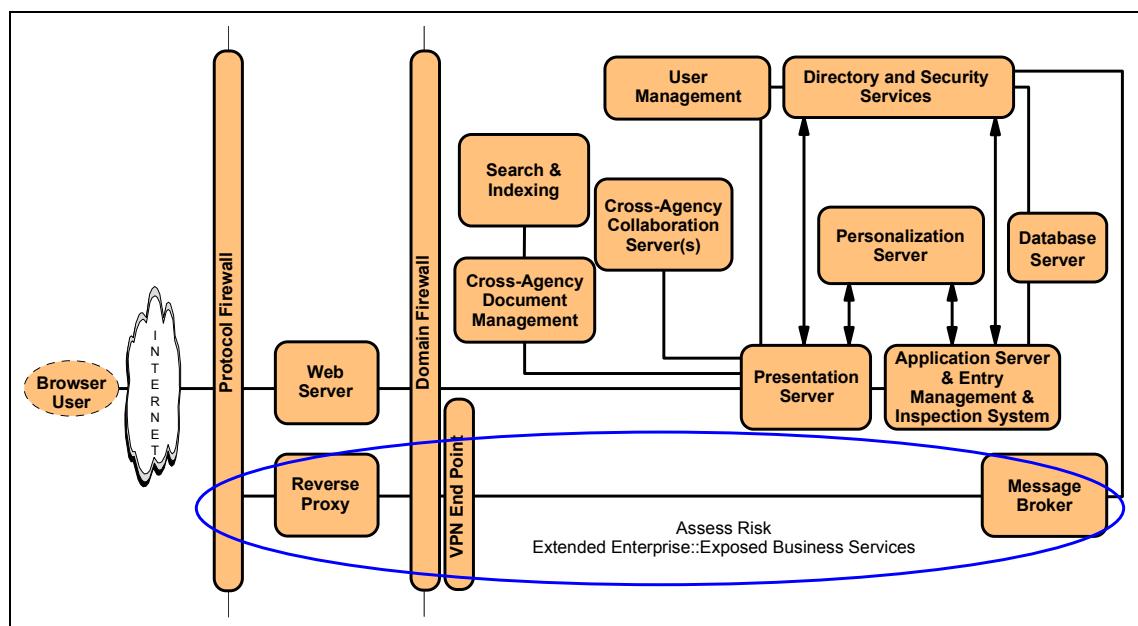


Figure 3-24 AOD (logical nodes) after the Runtime patterns were updated for the delta "Assess Risk"

Continue through the process described above to address the other deltas, updating the AOD (Logical Nodes) to reflect your choices. Figure 3-25 is the final AOD (Logical Nodes) after all deltas have been addressed.

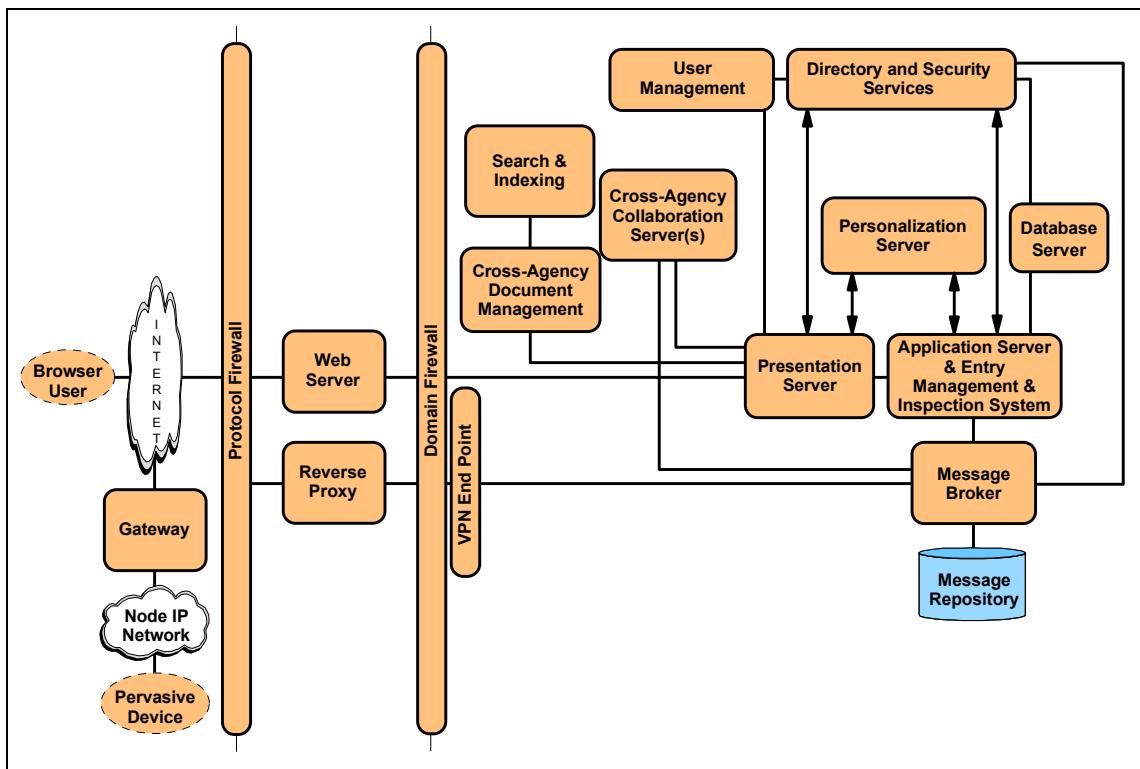


Figure 3-25 AOD (Logical Nodes) after all deltas have been addressed and the appropriate Runtime patterns incorporated

Table 3-19 details the updates that were made to the AOD (Logical Nodes) to address each of these deltas shown in Figure 3-25 on page 134.

Table 3-19 Description of how the deltas were addressed in the AOD (Logical Nodes)

Deltas identified in the AOD (subsystem)	Addressed by
Application Integration (Assess Risk)	<ul style="list-style-type: none"> ▶ <i>Applied Pattern:</i> Application Integration::Broker ▶ Added the Message Repository to address the Runtime pattern.
Access Integration (Inspect Cargo)	<ul style="list-style-type: none"> ▶ <i>Applied Pattern:</i> Access Integration::Pervasive Device Access ▶ Added the gateway and the non-IP network to address the Runtime pattern.
Extended Enterprise (Find Agency Information)	<ul style="list-style-type: none"> ▶ <i>Applied Pattern:</i> Extended Enterprise::Exposed Business Service ▶ Added the connection from the Message Broker to the Cross-Agency Collaboration Server to address the Runtime pattern. The reverse proxy, VPN end point, and Message Broker had already been applied when the Extended Enterprise (Assess Risk) delta was addressed in detail earlier.
<ul style="list-style-type: none"> ▶ Application Integration (Generate Alert) ▶ Application Integration (Generate Incident) 	<ul style="list-style-type: none"> ▶ <i>Applied Pattern:</i> Application Integration::Direct Connection ▶ No nodes/connections were added

Step 2D4: Capture decisions for chosen patterns into the Architecture Decisions document

Again, this formal task has been added to ensure that you understand the importance of documenting the decisions you have made during this process in the Architecture Decisions document. The tables below summarize the decisions that were made when the deltas were addressed in the preceding tasks above.

Subject Area	Inter-Agency Business Services
Architectural Decision	Extended Enterprise (Assess Risk)
Issue or Problem	Agencies need to protect their ability to maintain applications while preserving a consistent interface for consumption by partner agencies.
Assumptions	Since agencies will define their own private processes, support for a public process is not required

Subject Area	Inter-Agency Business Services
Motivation	Support good decoupling between agency systems and applications that should integrate.
Alternatives	
Decision	Applications are provided via the Exposed Business Service application pattern and will exploit Web services standards to manage the interface.
Justification	
Implications	Extensive use of Web services throughout e-government.
Derived Requirements	
Related Decisions	

Subject Area	Inter-Agency Business Services
Architectural Decision	Application Integration (Assess Risk)
Issue or Problem	This is a process (as opposed to data) integration activity.
Assumptions	The integration involves multiple systems.
Motivation	<p>Application pattern choices are</p> <ul style="list-style-type: none"> ▶ Direct Connection ¹ ▶ Transactional ² ▶ Broker ▶ Managed Process ³ <p>¹ Since this integration involves multiple systems, <i>Direct Connection</i> is inappropriate.</p> <p>² Since there are no updates to any of the contributing applications, this integration isn't <i>Transactional</i>.</p> <p>³ <i>Managed Process</i> (BPM) is typically associated with long running processes, whereas this is short-term.</p>
Alternatives	
Decision	Applications are provided via the Broker application pattern.
Justification	
Implications	
Derived Requirements	
Related Decisions	

Subject Area	Inter-Agency Business Services
Architectural Decision	Access Integration (Inspect Cargo)
Issue or Problem	Pervasive access to the system
Assumptions	
Motivation	This is an optional Portal composite pattern Application pattern driven by the <i>input device</i> characteristic.
Alternatives	
Decision	Applications are provided via the Pervasive Device Access application pattern.
Justification	
Implications	
Derived Requirements	
Related Decisions	

Subject Area	Inter-Agency Business Services
Architectural Decision	Extended Enterprise (Find Agency Information)
Issue or Problem	Similar issues to other Extended Enterprise patterns (detailed above)
Assumptions	
Motivation	Leverage existing Message Hub from other requirements.
Alternatives	
Decision	Applications are provided via the Exposed Business Process application pattern.
Justification	
Implications	
Derived Requirements	
Related Decisions	

Subject Area	Inter-Agency Business Services
Architectural Decision	<ul style="list-style-type: none"> ▶ Application Integration (Generate Alert) ▶ Application Integration (Generate Incident)
Issue or Problem	Providing simple, non-transactional integration to peer applications
Assumptions	
Motivation	These are Information Request (data integration). Options are Direct Connection or Aggregator. Aggregator requires multiple source applications/data sources. Upon further decision/realization that each application will be deployed on the same Application Server, we are easily able to provide direct communication.
Alternatives	
Decision	Applications are provided via the Exposed Business Process application pattern.
Justification	
Implications	
Derived Requirements	
Related Decisions	

Step 2E: Review and apply logical design guidelines to the AOD (Logical Nodes)

Throughout the solution development process, you should ensure that known guidelines and best practices are being followed. An excellent source for these guidelines is the appropriate patterns redbook. What you want to do at this step is review the current solution architecture to determine if it is consistent with these known guidelines and best practices.

For example, in *A Portal Composite Pattern Using WebSphere V4.1*⁴² you will find the following portal solution guidelines:

- ▶ Leverage a central user directory.
- ▶ Provide a single mechanism for intercepting user requests and passing security credentials to various applications and data sources.
- ▶ Provide a session or authentication time-out for a user's logged-in session.

⁴² Redbook SG24-6869-00, ISBN 0738427446

The Directory and Security Services node provides a central user directory. A single authentication point for user interactions can be provided by the Presentation Server, or later by a Security Proxy in the DMZ. Similarly, the Presentation Server or Security Proxy can provide the session time-out.

Here's one other guideline gained from field experience:

- ▶ Needing user provisioning means the likelihood of needing an enterprise security manager.

We have a requirement for user provisioning and have factored out User Management as a separate logical node in the AOD. We strongly suggest that you seriously consider an enterprise security manager within the solution and review this during physical design.

Step 2F: Conduct technical walkthroughs to ensure that functional requirements are met

Now that the AOD (Logical Node) has been developed, you should account for all functional requirements before continuing to the physical model. Using the AOD (Logical Nodes), you can conduct technical walkthroughs with the customer to validate that the functional requirements are met. Figure 3-26 on page 141 is a mockup of the AOD (Logical Nodes), detailing the functional flow of the use case *Request Entry*.

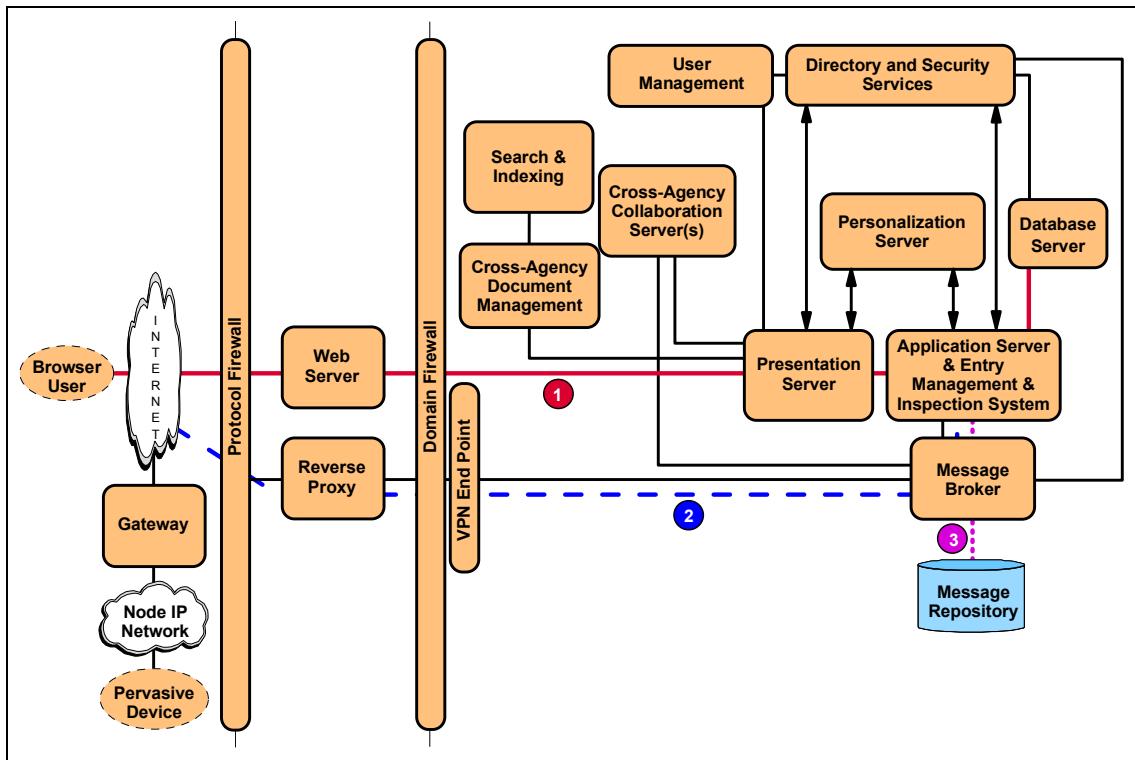


Figure 3-26 Mocked-up AOD (Logical Nodes) to be used in a technical walkthrough

The following details the steps highlighted in Figure 3-26.

Step 0

- ▶ Ship's captain logs into (authenticates with) the Port Authority portal.

Step 1

- ▶ Ship's captain navigates to the Entry Request Form via the portal and submits manifest and crew documents.
- ▶ The Presentation Server invokes the Entry Management application.
- ▶ The Entry Management application triggers the Assess Risk process in the integration hub.

Step 2

- ▶ The Assess Risk process accesses each remote business service via the Message Broker and VPN.
- ▶ The Assess Risk process determines that a alert must be generated.

Step 3

- ▶ The Assess Risk process sends an Alert via the integration hub and message repository to the Customs Agency.

Once you have verified that the logical solution you have prepared addresses all functional requirements, you can continue on to develop the physical solution.

3.2.3 Step 3: Develop the physical solution

At this point, you have finished designing the logical representation of your solution. You have (long ago) worked with the customer to determine the business model it was originally using. You have investigated its use cases, determined NFRs, made some architectural decisions, and determined which Business, Integration, Composite, and Application patterns were applicable to your design. Now it's time to develop the physical solution.

The physical system design is driven by the logical solution, developed using both the Component (optional) and Operational models, and serves to determine the actual product mappings for the solution.

Steps required to develop the Physical Solution

- ▶ Step 3A [optional]: Develop high-level component model
- ▶ Step 3B [optional]: Apply Design Guidelines (Application Level) from large reusable assets (including redbooks and best practice white papers)
- ▶ Step 3C [optional]: Perform a technical walkthrough of the component interactions
- ▶ Step 3D: Develop the Operational Model
- ▶ Step 3E: Seed the Operational Model by applying the Product mappings from the matched large reusable assets
- ▶ Step 3F: Address “operational” NFRs in the Operational Model
- ▶ Step 3G: Conduct technical walkthroughs to ensure that Non-Functional Requirements are met

This section addresses each of these steps.

Step 3A [optional]: Develop high-level component model

The Component Model describes the high-level logical structure of the system's components. It depicts major subsystems and boundaries (interfaces) of the overall system and:

- ▶ Shows and defines the components (nodes) that comprise the solution and are to be placed on the operational model

- ▶ Exhibits the responsibilities, relationship boundaries, and interactions of the components
- ▶ Documents how functional/technical parts of the system are related

A *component* can be a software subsystem, a program module, a collection of classes (for example, all the classes dealing with registration), a program (for example, one that performs alert notification), a part of a product (for example, DB2®), or a hardware device (for example, a computer).

The Component Model encompasses a number of diagrams. At the highest level, it identifies the components or nodes that make up the solution and shows how they interact. There may be a description of each of the components involved. Relationship diagrams show how the functions work together, and sequence diagrams exhibit what happens when, so that the process or data flow can be established.

Note: This work product can be considered optional if the solution is not complex enough to warrant its necessity.

Table 3-20 shows the pertinent components in the e-government case study, organized by associated logical node.

Table 3-20 e-government case study components

Logical Node	Associated System Components
Presentation Server	<ul style="list-style-type: none"> ▶ Health Department Portal ▶ Port Authority Portal
Cross-Agency Document Management	<ul style="list-style-type: none"> ▶ Content Management System ▶ Remote Document Search Service
Application Server, Entry Management, and Inspection System	<ul style="list-style-type: none"> ▶ Entry Management ▶ Customs Inspection System
Message Broker and Integration Hub	<ul style="list-style-type: none"> ▶ Alert Management ▶ Integration Hub ▶ Risk Assessment System
Cross-Agency Collaboration Server	Emergency Response Team Environment
Existing Agency Servers	<ul style="list-style-type: none"> ▶ Lookout Operational Data Store (ODS) Service ▶ Import Bill of Lading (BOL) Service ▶ Threat Level Service
Component added for performance purposes	Subject Matter Expert ODS

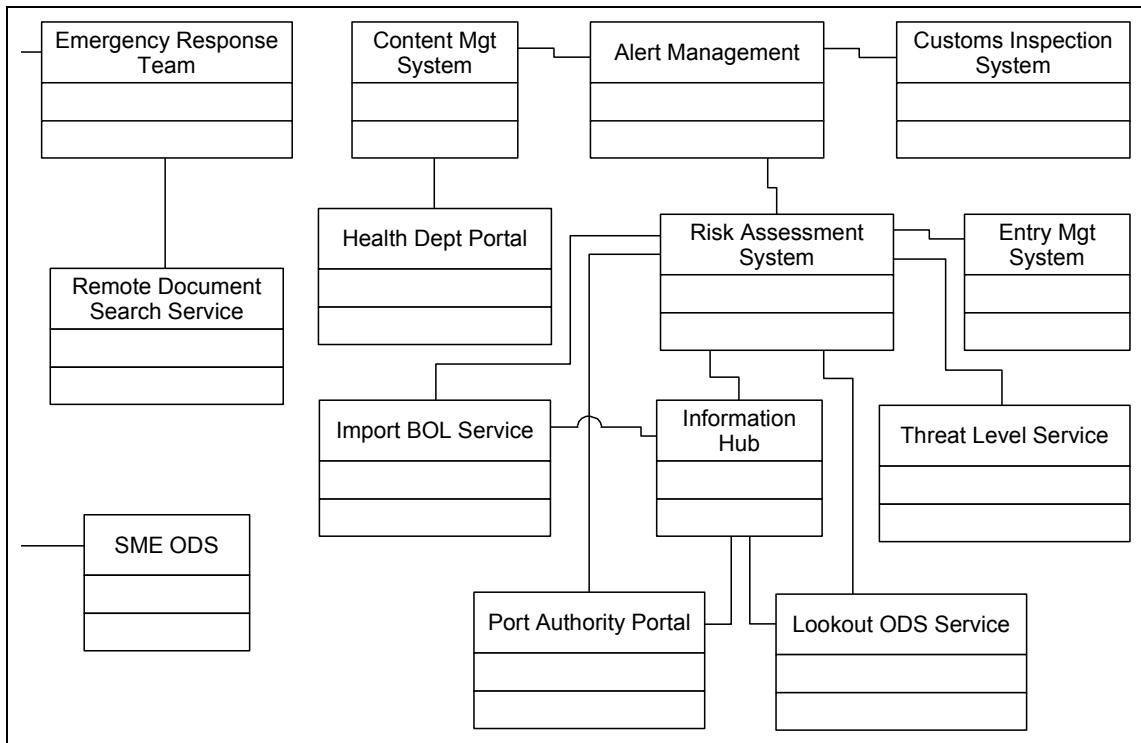


Figure 3-27 e-government case study Component Model Relationship Diagram

Figure 3-27 illustrates the relationship between the components that comprise your solution. Cardinality⁴³ can be noted if desired. The intent is to show which components interact with others.

Note: The output of this step is the Component Model Relationship Diagram work product.

Component Interaction Diagram for the case study using the Risk Assessment scenario

Component Interaction diagrams serve to validate your design in scenario context. They allow you to walk through the scenario and ensure that the use cases and components previously identified are correct with respect to their interactions.

⁴³ The number of cardinal (basic) members in a set. In tables, the number of rows is called the cardinality

Note: In the Risk Assessment and Inspection diagrams that follow (Scenarios 1-4), the wireless entity isn't really a component but is included in the diagram to enhance readability and understanding. Also note that although no actual storage is shown, a record is created and stored for the Risk Assessment scenario. This record can be referenced later for the inspection.

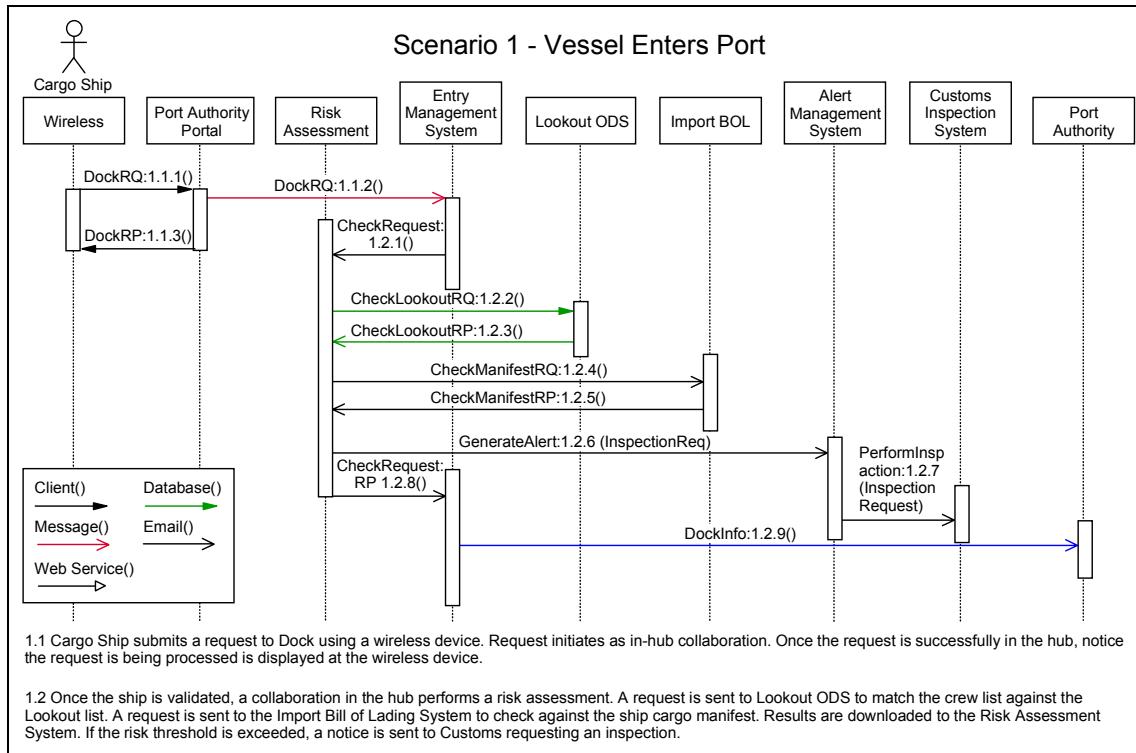


Figure 3-28 e-government case study Component Interaction diagram—Scenario 1

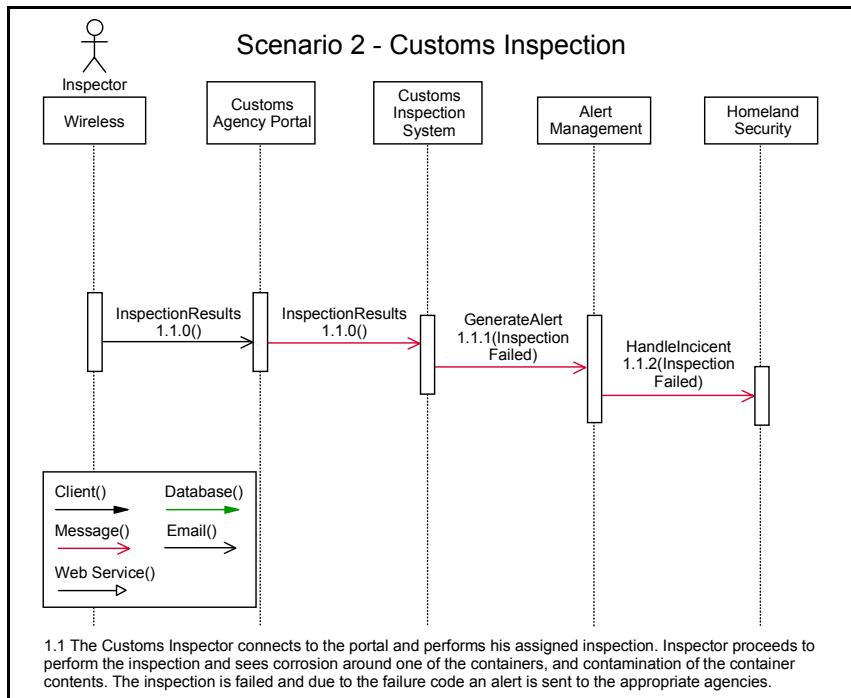


Figure 3-29 e-government case study Component Interaction diagram—Scenario 2

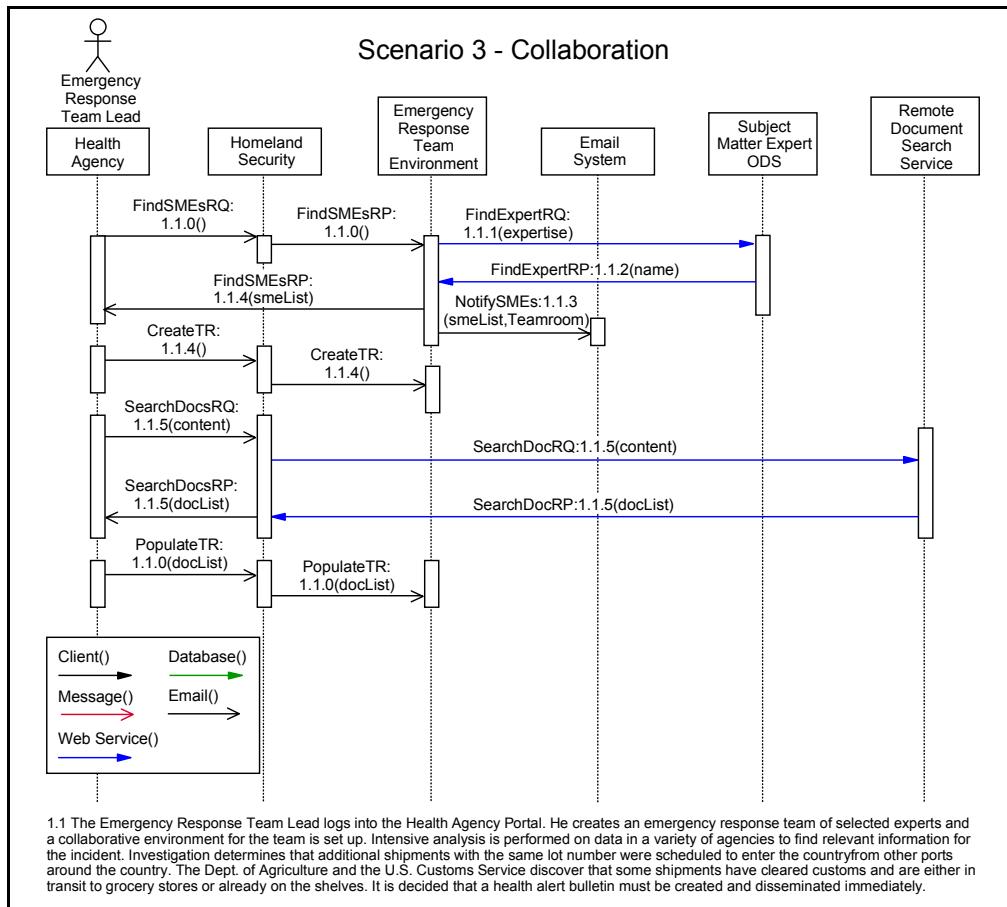


Figure 3-30 e-government case study Component Interaction diagram—Scenario 3

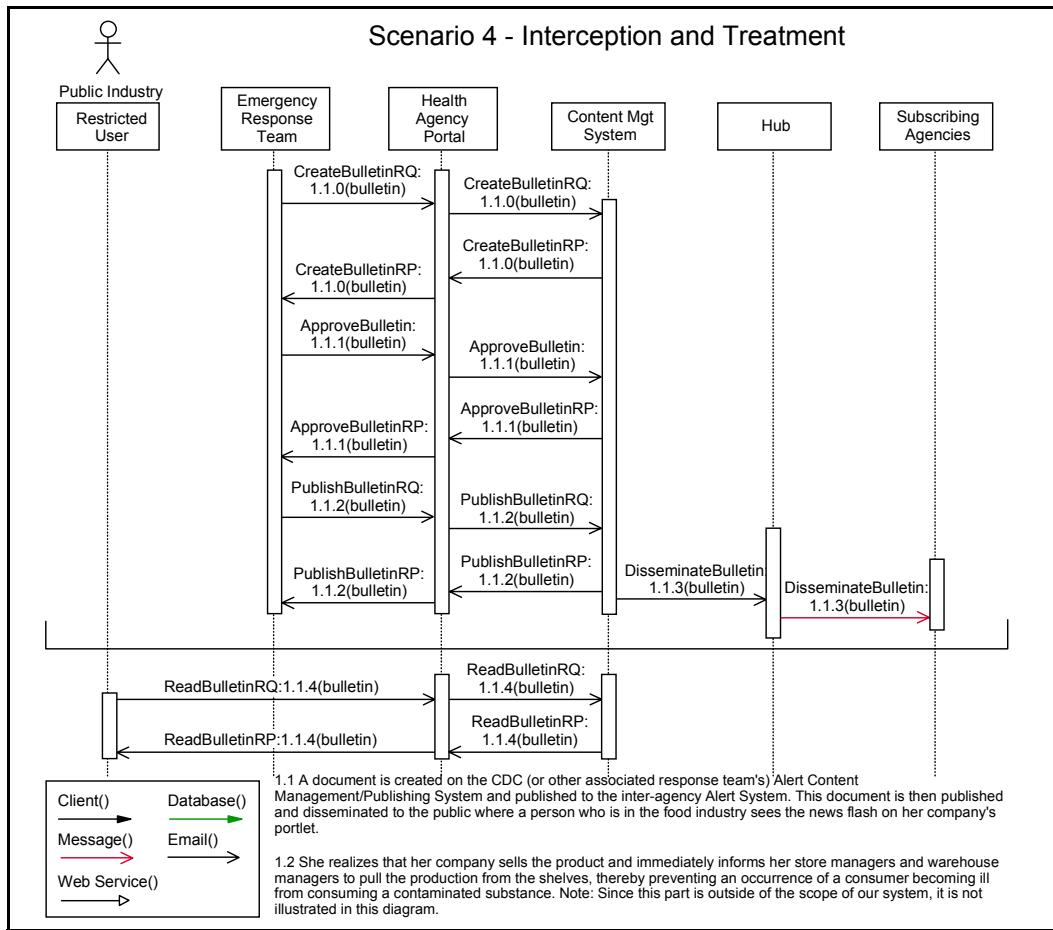


Figure 3-31 e-government case study Component Interaction diagram—Scenario 4

Step 3B [optional]: Apply Design Guidelines (Application Level) from large reusable assets (including redbooks and best practice white papers)

Guidelines help establish the required service levels needed to support functional responsibilities. Their use helps ensure that these responsibilities are covered by the set of components representing the solution architecture. Refer to the guidelines published in relevant patterns redbooks. For example, review Chapter 8, “Application Design,” from *A Portal Composite Pattern Using WebSphere V4.1*.⁴⁴ There you will find guidelines on topics such as single-versus multi-tier Portal server design, content management, MVC, SSO, collaboration, and Web services guidelines. Chapters 7-10 provide a set of guidelines organized by technology, application design, application development, and systems management.

Content Management guidelines included in this redbook:

- ▶ Define content types and location/source
- ▶ Define the process by which content will be contributed and eventually published
- ▶ Define the content versioning and expiration schemes

SSO usage guidelines and best practices from this redbook

- ▶ Leverage a central user directory
- ▶ Provide a single mechanism for intercepting user requests and passing security credentials to various applications and data sources
- ▶ Provide a session or authentication timeout for a user's logged-in session
- ▶ Provide a variation of the SSO interface for various client device types
- ▶ Identify the various user types and link them with the business rules engine
- ▶ Enable encryption of the data packets as authentication is being performed
- ▶ Update security policies to enhance the physical security of the SSO mechanism that intercepts authentication requests for existing applications and data sources
- ▶ Agree to a common format for user names and passwords
- ▶ Agree to a process for updating SSO account information for user names and passwords⁴⁵

In addition to the redbook guidelines, be sure to leverage physical mappings from known solutions and field experience, both for core components and for key component interactions.

⁴⁴ Redbook SG24-6869-00, ISBN 0738427446

⁴⁵ For example, passwords cannot be retrieved if lost, but must be reset

Once the guidelines have been found and identified, the question is how to apply them for practical use. For example, the Content Management guidelines previously specified imply a need for an Authoring server with workflow. Also, a decision can be made as to how the content will be delivered (in other words, Runtime server). These may be new, additional components, or additional details for existing components, and they will be reflected in the Operational Model.

Step 3C [optional]: Perform a technical walkthrough of the component interactions

The main idea is to verify that the component interactions and sequences apply to the use cases. This is to insure that the operational flow is correct and that all components are accounted for, and should be very detailed. The Component Interaction and System Sequence diagrams are used to document and validate system behavior. The walkthrough should ensure coverage of all system scenarios (detailed use cases) by interaction or system sequence diagrams.

Develop test case scenarios from the use cases to ensure that all are correct. Read the text insert example in the Component Interaction diagram used for our case study and verify that it validates the use (and test) case.

Step 3D: Develop the Operational Model

After reviewing the AOD, the next logical step is to translate the logical nodes into physical servers and then apply the products needed to complete the physical model. This is accomplished by starting with the AOD to seed the Operational Model (which is a graphical representation of a network of computer systems, their associated peripherals, and the systems software, middleware, and application software that they run). It answers such questions as how many servers and where components reside within them.

The Operational Model serves to:

- ▶ Provide a physical view of the system
- ▶ Depict major elements of the IT system
- ▶ Facilitate analysis of NFRs
- ▶ Provide a walkthroughs mechanism
- ▶ Enable product evaluation and selection
- ▶ Enable early cost estimates

It may contain:

- ▶ System Topology Diagram
- ▶ Node descriptions
- ▶ Network view
- ▶ Software view
- ▶ Physical hardware view

► Walkthrough description

As previously stated, certain work products are referenced when creating others. When creating the Operational Model, be sure to refer to the NFRs, Architectural Decisions, Architecture Overview Diagram (Logical Nodes), and Component Model (which gets deployed onto the Operational Model). These work products will assist you in developing your Operational Model. Also see if there are any design guidelines from associated redbooks that are applicable. Use field expertise when considering clustering, etc.

The process of creating the Operational Model is illustrated in Figure 3-32.

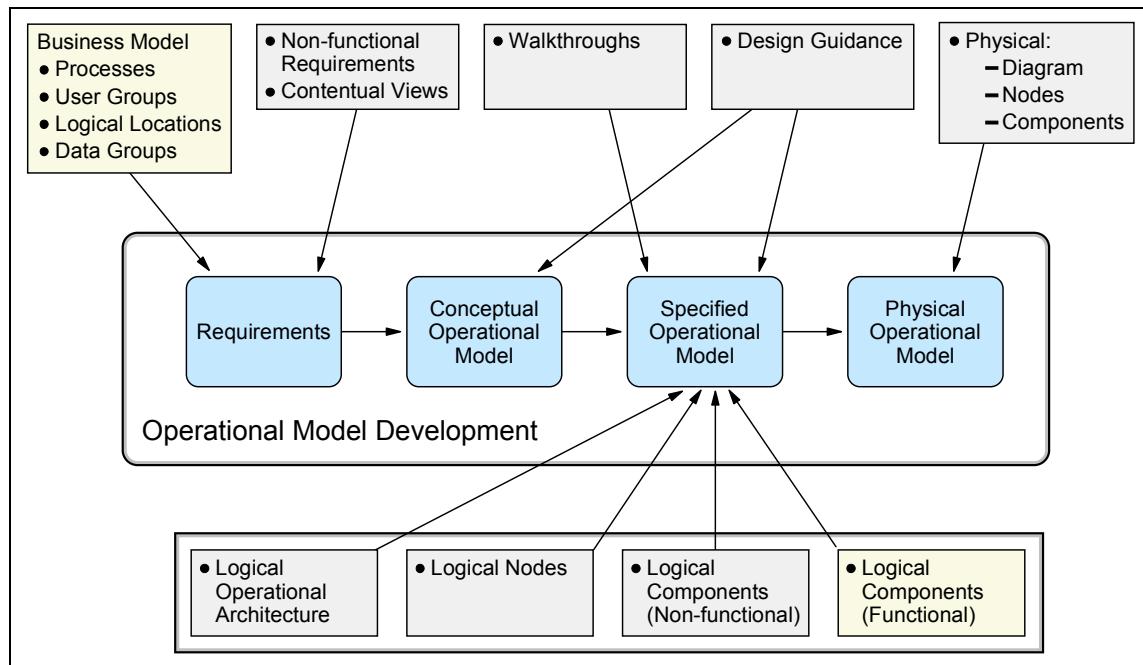


Figure 3-32 Operational Model development process

The Conceptual Operational Model references project requirements. The Specified Operational Model references the logical components, and the Physical Operational Model references the physical components. Generally, an Operational Model develops from *conceptual* to *specified* to *physical*. Depending on the complexity of the problem and your starting point, it may not be necessary to go through all three stages. For example, an architecture may be heavily constrained by physical platform decisions that have already been made, or by an existing specification-level known solution. Remember to address the NFRs (for example, clustering or high-availability) and any product dependencies in your Operational Model.

First we start with the AOD for our case study (as shown in Figure 3-33 on page 153). Note the additional documentation of the software services provided by the logical nodes. This will help provide a foundation for establishing the product mappings that follow directly afterward.

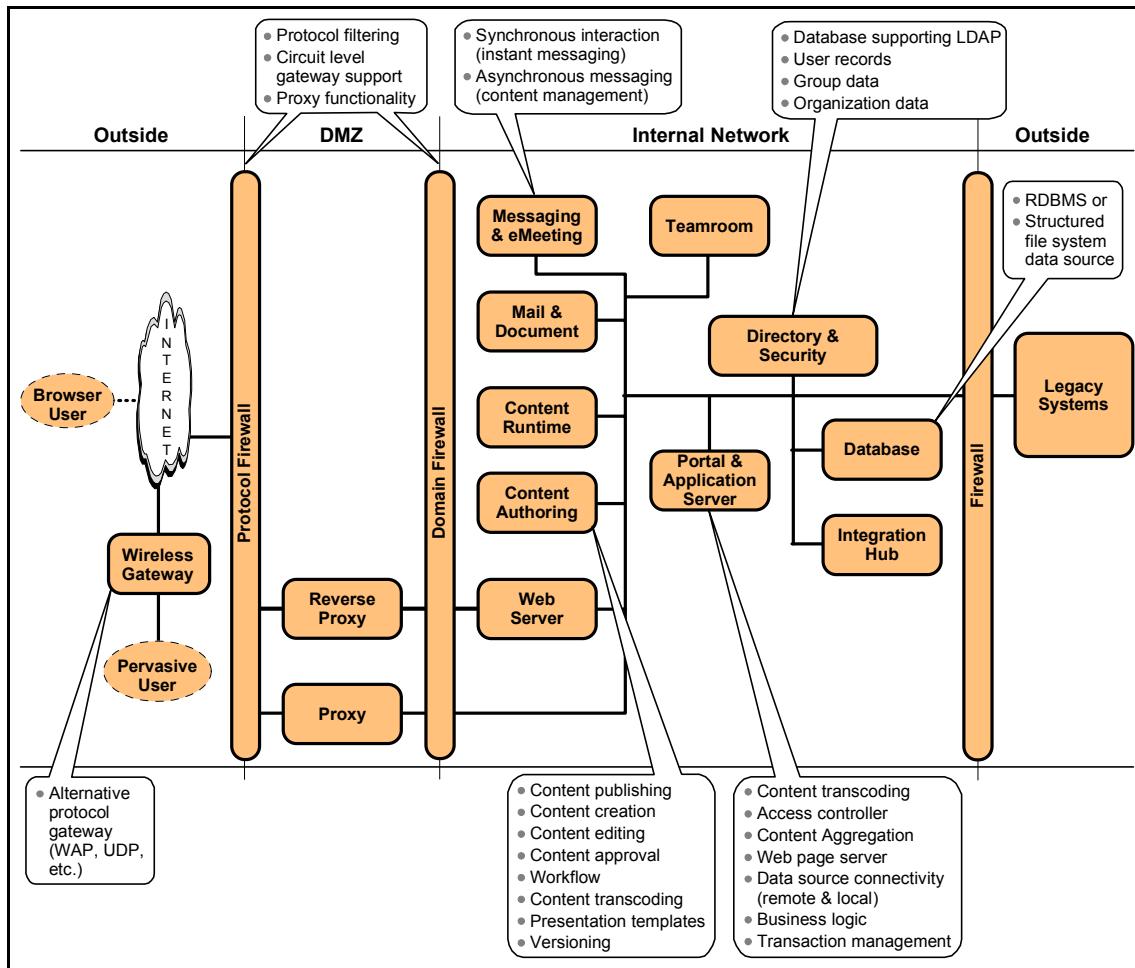


Figure 3-33 AOD for the e-government case study

Step 3E: Seed the Operational Model by applying the Product mappings from the matched large reusable assets

Apply product mappings for the delta requirements—Unfortunately, solutions architects often jump right into this step without carefully considering all the previous steps and work products that we have discussed so far in this chapter. The problem with this approach is that it makes it harder to maintain and scale the proposed solution. It becomes more of a “standalone” architecture that may work for now, but probably not last for the long term. Or it may not integrate easily into the customer's existing IT infrastructure.

To properly perform product mapping, it's essential to go through the method outlined in this chapter—to go through the steps and create all of the associated work products. It's a best practice to identify the Business, Integration, Composite, Application, and Runtime patterns in order to arrive at the point where product mappings can occur. From the Runtime patterns identified in your solution, you can map the products, technologies, and platforms that provide a "best fit" for your solution. Realize, of course, that you should also leverage any existing assets. Use your collection of assets⁴⁶ to identify the best implementation approach for your components.

In addition to business drivers, consider the following when deciding on a product and technology mix:

- ▶ *Existing systems and platform investments*—Leverage these as much as possible unless there is a reason to change.
- ▶ *Customer choices*—The customer might have experience and satisfaction with certain products and operating systems and thus want to stay with them.
- ▶ *Planned future functional enhancements*—Try to obtain a "long-range" vision based on the customer's goals. Look at the Envisioned Goals and Issues document. Is there anything in there suggesting that certain topologies or products should be adopted at this time? Maybe the customer has plans to scale up at a later time. You must consider this so as not to inhibit scalability at a later date with a product set that limits this capability.

The products and technologies chosen should fit into the target environment and ensure quality of service (such as scalability and reliability) so that the solution can grow along with the e-business.

The goal in product mapping is to choose a set of products and technologies that will minimize any necessary customization. Refer to any known solutions that are applicable and show products and versions that are known to work well together. Realize, however, that product updates may have occurred since that architecture was deployed. Be sure to check the product release notes⁴⁷ for compatibility listings before making any final decisions. There might be version dependencies that must be observed. For example, when integrating TAM V4.1 and WebSphere Portal Server V4.2, there is a portal server patch (PQ70837) that must be applied—a patch that isn't needed when using WebSphere Portal Server 4.2.1.

Unless this is a stand-alone solution or there is no current target environment, there may be an existing set of products and technologies that have already been implemented for specific functionalities. In this situation, the integration of products and technologies from various vendors (for example, IBM, Microsoft,

⁴⁶ Known solutions, Composite patterns, field deployments, etc.

⁴⁷ <http://www.ibm.com/software/support>

Sun, and Oracle) is not only required but can be successfully implemented if the products communicate—for example, by using Web services. If you need to use existing product sets, this will affect your architectural decisions. Be sure to document these.

The Patterns for e-business Web site⁴⁸ shows each Runtime pattern with the products that have already been tested in that capacity. It may be beneficial to reference these because they have already been proven to be successful. Note, however that this should be used as a starting point, that it is not exhaustive and may not always represent the product versions that are appropriate for your particular engagement. A more successful approach might be to leverage existing known architectures. In addition, be sure to review Chapter 8 of *A Portal Composite Pattern Using WebSphere V4.1*.⁴⁹ This chapter includes guidelines for SSO, collaboration, Web services, and portal solutions, all of which should be considered before any product mapping occurs.

From this, we derived the product mappings. In other words, we “filled in” or listed the products that provided the necessary functions for each node, resulting in the design shown in Figure 3-34 on page 156.

⁴⁸ <http://www.ibm.com/developerworks/patterns/>

⁴⁹ Redbook SG24-6869-00 (ISBN 0738427446)

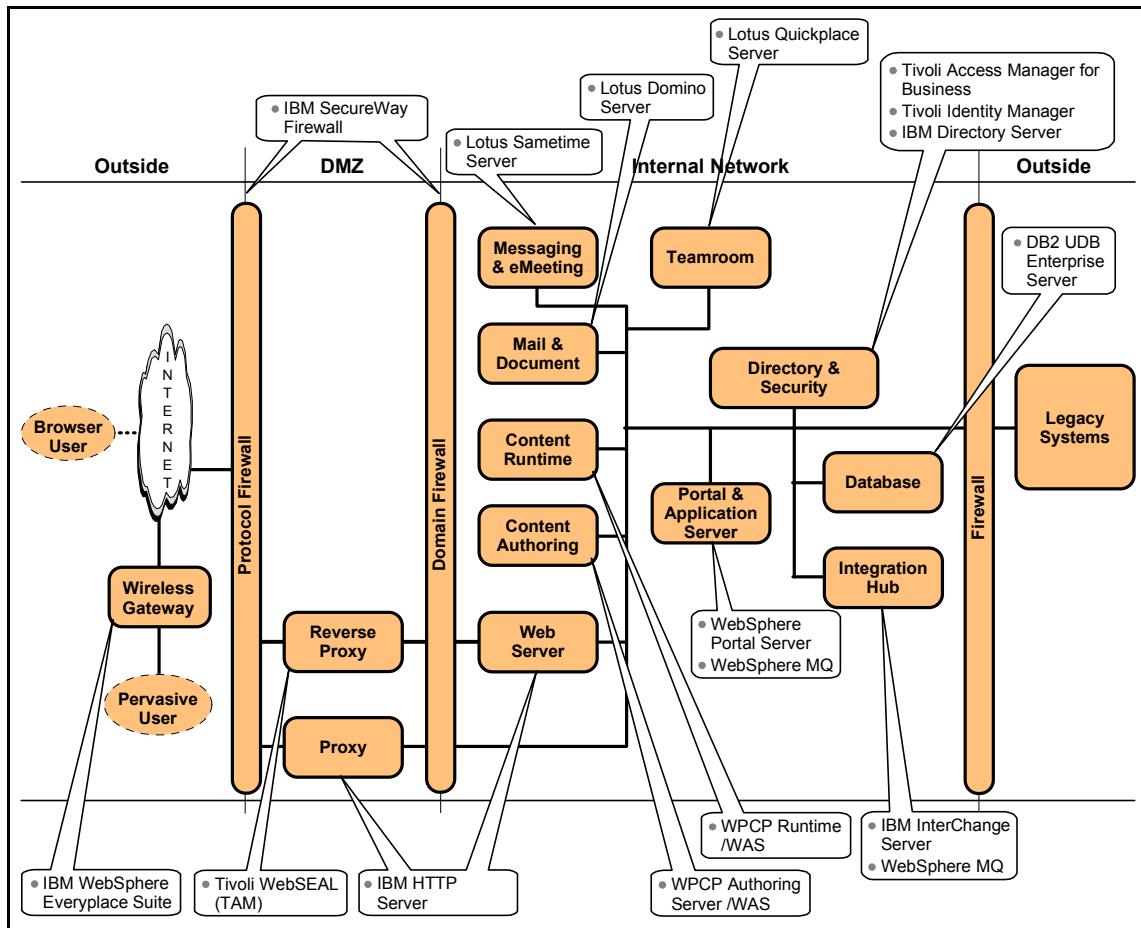


Figure 3-34 AOD for e-government case study with product mapping

From this diagram we derived the Operational Model illustrated by Figure 3-35 on page 157.

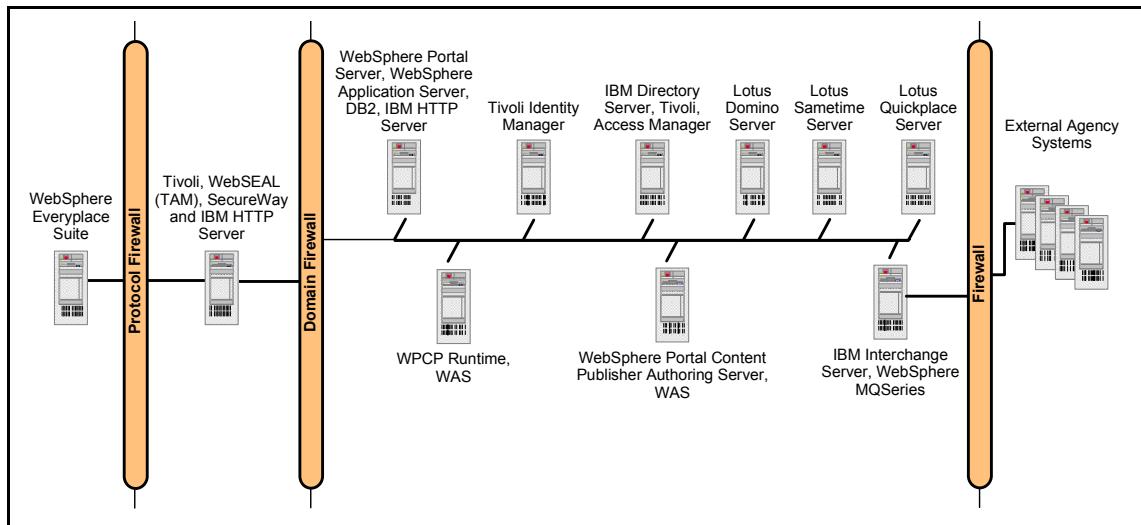


Figure 3-35 e-government case study Operational Model

This gave us the base model. Then—by referring to the NFRs to see what performance, scaling, and high-availability requirements were needed—we added additional servers to address these requirements (as discussed in the next task).

Note: The output of this step is the Operational Model work product.

Step 3F: Address “operational” NFRs in the Operational Model

At this point, you have a first pass at your physical model. You have applied the product mappings, taking advantage of any available known solutions, redbooks, and other guidelines and best practices. Now it's time to address the NFRs in your physical model to ensure that they are all accounted for. NFRs are used to assess the viability of the proposed IT system. *NFRs are frequently the most important determining factor of the architectures.*

IBMers should realize that this is another opportunity to use redbook guidelines, physical mappings from known solutions, and field experience. If you have any other engagements that concerned themselves with these issues, you should examine what was done to address these concerns. You might be able to leverage these assets. Also leverage the IBM High Volume Web Site (HVWS⁵⁰). The HVWS team focuses on high scalability tests and measurements, end-to-end performance management, and collaboration with key product groups for high-scale testing in customer environments. From this site you can

⁵⁰ <http://www7b.software.ibm.com/wsdd/zones/hvws/>

find best practices, actual design and implementation experiences, and presentations on scalability and performance that might assist you in your solution. These were generated from actual customer engagements. There are many guidelines and white papers dealing with scalability and performance. Following is an example for addressing *high availability, security, and performance* concerns from the HVWS:

- ▶ Cluster the portal server (implies clustering the underlying applications server).
- ▶ Cluster the Web server to enhance delivery of static content.
- ▶ Use WebSEAL as a reverse proxy to keep security mapping (LTPA tokens) in the DMZ. Replicate/cluster the WebSEAL server to enhance throughput.
- ▶ Cluster/replicate other key-service intensive servers based on expected traffic (for example, various collaboration servers such as Lotus® Instant Messaging and Team Workplace®).⁵¹

Using these clustering guidelines, our Operational Model from Figure 3-35 on page 157 evolves into Figure 3-36.

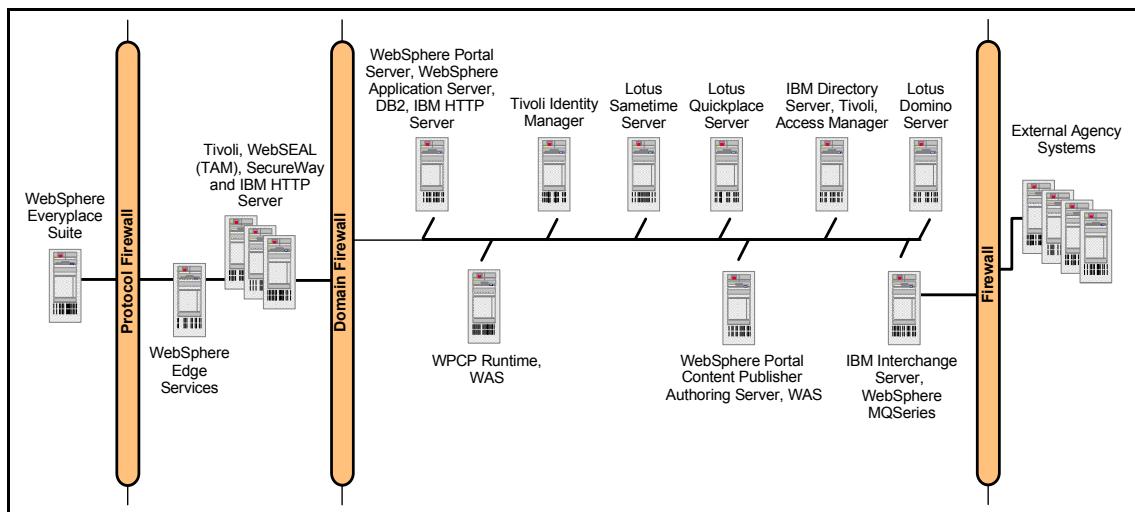


Figure 3-36 Operational Model with clustering

Another example highlights best practices for achieving application scalability.

⁵¹ Formerly known as Sametime and QuickPlace

Ten techniques for achieving scalability

1. Faster machines

The goal is to increase the ability to do more work in a unit of time by processing tasks more rapidly. A faster machine can be achieved by upgrading its hardware or software.

2. Replicated machines

The primary goal is to service more client requests. Parallelism in machine clusters typically leads to improvements in response time. Moreover, system availability is improved due to failover safety in replicas.

3. Specialized machines

The goal is to improve the efficiency of a specific component by using a special-purpose machine to perform a required action. These are usually dedicated machines that are very fast and optimized for a specific function. Examples are network appliances and routers with cache, such as the IBM WebSphere Edge Server.

4. Segmented workloads

The goal is to split up the workload into manageable chunks, thereby obtaining more consistent and predictable response time. The technique also makes it easier to manage workload distribution on individual servers. Combining segmentation with replication frequently offers the added benefits of providing an easy mechanism for redistributing work and scaling selectively as business needs dictate.

5. Request batching

The goal is to reduce the number of requests sent between requesters and responders (such as between tiers or processes). This is accomplished by defining new requests that combine multiple requests.

6. User data aggregation

The goal is to allow rapid access to large customer data controlled by existing system applications and to support personalization based on customer-specific data. The technique calls for aggregating customer data into a *Customer Information Service* (CIS). A CIS that is kept current can provide rapid access to customer data for a very large number of customers, thereby providing the required scalability.

7. Connection management

The goal is to minimize the number of connections needed for an end-to-end system, as well as to eliminate the overhead of setting up the connections. This is accomplished by maintaining and sharing a pool of preestablished connections that can be reused. In WebSphere, this is referred to as *connection pooling*. Reusing existing database connections conserves

resources and reduces latency for application requests, thereby helping to increase the number of concurrent requests that can be processed. Managing connections properly can improve scalability and response time.

8. Caching

Caching is a key technique to reduce hardware and administrative costs and to improve response time. The goal is to improve performance and scalability by reducing the length of the path traversed by a request and the resulting response, and by reducing the consumption of resources by components when the same content is requested multiple times or by multiple users.

9. Content delivery services

The goal is to guarantee that mission-critical content (video, audio, data, or text) gets from its source to its destination in a timely and reliable fashion. One method is to distribute the servers in multiple localities, thereby ensuring that content can be provided even if one server is down, for load balancing, and for delivering from the best possible location for enhanced performance.

10. Local distribution across sites

The goal is to distribute network traffic across multiple sites, thereby reducing the load and bottleneck at any one site. Locally distributing the application across multiple sites enhances scalability.

Table 3-21 How scaling techniques relate to scaling objectives

Technique #	Scaling Technique	Increase Capacity/Speed	Improve Efficiency	Shift/Reduce Load
1	Use faster machine	X		
2	Create machine cluster	X		
3	Use a special machine	X	X	
4	Segment the workload		X	X
5	Batch requests		X	
6	Aggregate user data		X	
7	Minimize connections		X	
8	Cache data and requests		X	X
9	Content Delivery Services	X		X
10	Local distribution across sites		X	X

Figure 3-37 illustrates where you might take advantage of the preceding scalability guideline techniques.

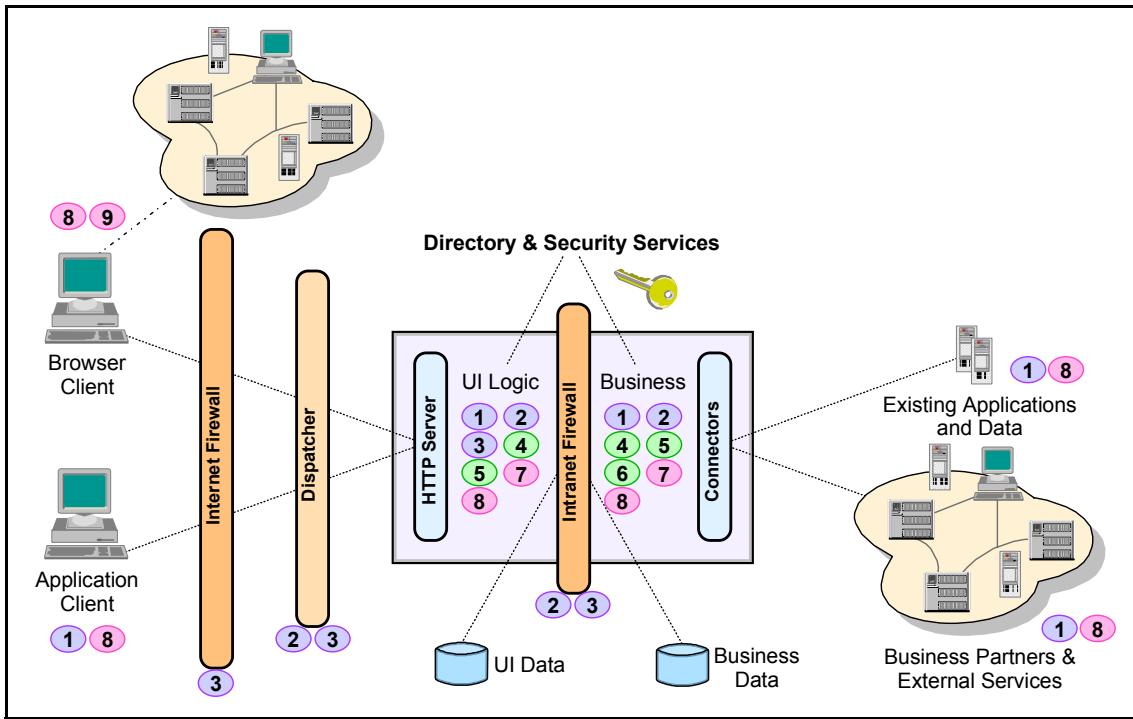


Figure 3-37 Where performance, scalability, and availability can be implemented

It's important to note that a detailed understanding of the application is key to the successful implementation of these best practices in an architecture.

Note: The IBM High Volume Web Site is
<http://www.ibm.com/websphere/developer/zones/hvws/>

Step 3G: Conduct technical walkthroughs to ensure that Non-Functional Requirements are met

Technical walkthroughs need to consider all applicable NFRs to ensure that availability and performance are covered. You should also consider maintenance

requirements (for example, can regular maintenance be applied within the availability guidelines?).

1. For our first walkthrough, we'll use the high-availability NFR that was addressed in Step 1F: Document Non-Functional Requirements.
 - Configurable to achieve 24x7 but not required in the first installation
- Refer to Figure 3-38 on page 162. Let's assume that the requirement is that any user must be able to connect to the portal server at any time (in other words, that it's always available).

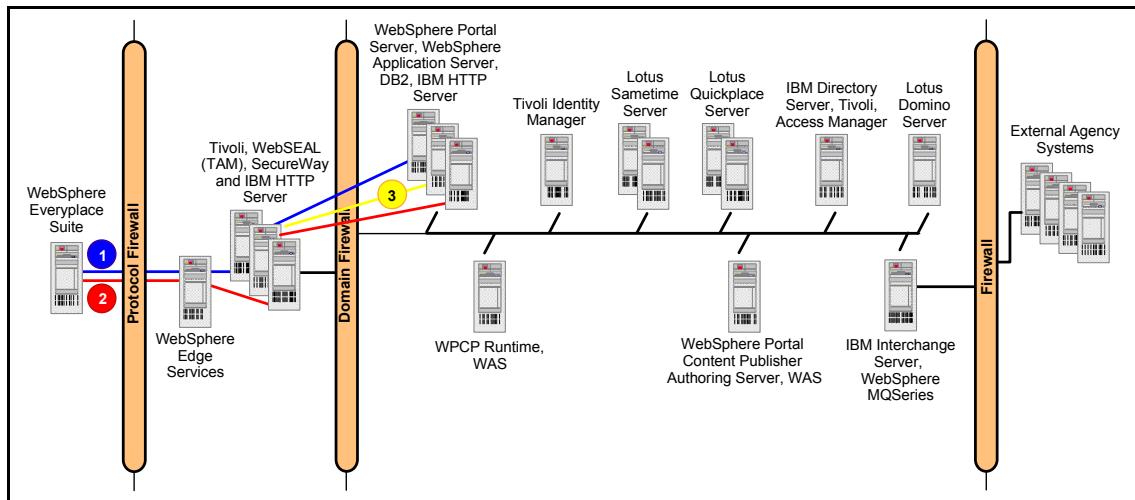


Figure 3-38 e-government case study Operational Model with high-availability NFR walkthrough

Detailed below are the walkthrough steps that are highlighted in Figure 3-38 to ensure that this requirement is met.

Note: The colored lines and numbered circles correspond to users, not the step numbers.

- **Step 1:**
User “1” logs into the portal server (**blue** path)
- **Step 2:**
User “2” logs into the portal server (**red** path)
- **Step 3:**
The portal server that User “1” is logged into fails over to another in the cluster (**yellow** path). User “1” continues uninterrupted.

2. For our next walkthrough, we'll use the performance-related NFR that was addressed in Step 1F: Document Non-Functional Requirements (see Figure 3-38).
 - Concurrent access from multiple users to be supported.⁵²

Again using Figure 3-38, you can see that the security servers between the two left-most firewalls are redundant. This is for failover as well as for load balancing to increase performance. Notice that the portal server as well as the Lotus Instant Messaging and Team Workplace servers are all redundant—which addresses this particular NFR as well. We can use a similar walkthrough as the one above.

- **Step 1:**

User “1” logs into the portal server (**blue** path)

- **Step 2:**

User “2” logs into the portal server (**red** path)

Additional users that log in will be load-balanced among all available redundant servers.

3. For our next walkthrough, we'll use the security-related NFRs that were addressed in Step 1F: Document Non-Functional Requirements. These are handled by the WebSEAL and Tivoli servers and products. As a reminder, the following NFRs address security.

- **Authentication**

- **NFR 1.** Access validation by user ID/password will be provided.
- **NFR 2.** Usage of passwords for supervisor functions will be provided.
- **NFR 3.** System administrator authority to set up and maintain application security for functions and data will be provided.
- **NFR 4.** An interface for use by server software to authenticate clients will be provided.

- **Authorization**

- **NFR 5.** Validation of authorized clients will be provided by server components.

- **Other**

- **NFR 6.** Passwords will be masked/not printed at entry.
- **NFR 7.** Passwords will be stored in encrypted format.
- **NFR 8.** PKI or Basic authentication/authorization
- **NFR 9.** LDAP based

⁵² Expect 20,000 agency users in addition the list of external registered users.

The walkthrough is as follows:

– **Step 1:**

User attempts login to the WebSEAL server (NFR 4) through a browser.
The user is challenged to provide user name and password (NFR 1).
Passwords are masked as they are typed (NFR 6).

– **Step 2:**

WebSEAL authenticates the user (NFR 5). This is performed by passing the user credentials to Tivoli Access Manager (TAM), which checks the credentials and encrypted password (NFR 7) against the LDAP user registry (NFR 9) and returns to WebSEAL.

– **Step 3:**

If authentication is validated, WebSEAL forwards the HTTP request and authentication credentials into the trusted network to the portal server.

– **Step 4:**

The portal server uses the Trusted Association Interceptor (TAI) to communicate with Tivoli Access Manager (TAM) to verify that the user is authenticated. TAM then verifies user authorization.

– **Step 5:**

If authorized, the portal server displays the requested URL.

– **Step 6:**

The portal session is started and an LTPA token created and cached in WebSEAL's memory so that at the next user request, WebSEAL determines that the token already exists and passes the request to the portal server without repeating the authentication. The portal server checks the TAI and displays the page.

For SSO, the token is passed to the other servers when they are accessed so that the user will not be challenged a second time.

Assumptions: WebSphere Portal Server is configured to use an external authentication proxy. WebSphere Portal Server has deferred ACL to TAM. Authentication is HTTP Basic Authentication (NFR 8).

For brevity, we will not address each of the NFRs in this chapter, but this is something you will need to do to ensure that your solution takes them all into account. Should you find that you overlooked one of the NFRs, this is the time to address that. For instance, if you see that there's an NFR for high availability and you haven't already designed, this is the time to add it to your physical model.

3.3 Implement

By now you've engaged the customer to gather requirements, issues, and future goals. You've analyzed its current IT infrastructure and business model to gain an understanding of how its business operates today, and you've documented a solution architecture and created all of the related diagrams and associated documents.

At this point, it's time to obtain customer approval of your solution. This is the *Implement* phase, and it's here that you'll review it with the customer to make refinements and resolve any concerns still outstanding. Once this has been accomplished, you should review the guidelines on the Patterns for e-business Web site⁵³ for each of the application patterns selected. These guidelines provide tips and techniques for designing and developing the application with performance and system management in mind.

Once the pilot application has been developed, it needs to go through extensive testing and monitoring. Go back to the Critical Success Factors that you documented in the Envisioned Goals and Issues document and see if these are being met. This is how success is measured. Ensure that the customer is satisfied with the solution.

Once the engagement is complete, it's time to harvest assets and make them available for future engagements. You now have a known solution and implementation that's available for reuse.

⁵³ <http://www.ibm.com/developerworks/patterns/>



Architecting On Demand Workplace solutions

In this chapter we will work through a second case study to further illustrate the development process presented in Chapter 3. The case study presented here focuses on a Business-to-Employee (B2E) portal. Such portals have applicability in any industry setting where organizations focus on creating streamlined, more fluid, more responsive and productive working environments for their employees. The goal is to make everyone a “knowledge worker” in an organization where individuals can use self-service human resource (HR) systems, boost collaboration, communicate anywhere, and otherwise relate to one another and to the company in a whole new way.

These knowledge workers should have a single point of access to the information and tools they need *when they need it*. Such a site aligns with IBM's On Demand Workplace (ODW). This case study explores some of the common elements associated with any ODW solution. At the end of the chapter we will see how a generic large reusable asset can be derived for this broad class of solutions.

4.1 Retail case study introduction

For this activity, we will build a Portal solution for International Food and Beverage (IFB), a retail customer,. What follows is a solution description that has been documented after an initial set of meetings with the customer.

4.1.1 Company background

IFB is a leading worldwide restaurant chain, with annual sales of around US\$ 4 billion, 30,000 employees, and nearly 480 restaurants. It was originally a UK-owned company that grew to a multinational corporation with restaurants throughout Europe, Asia Pacific, and North America. It had a workforce of 10,000 full-time and 20,000 part-time employees spread across the corporate offices in Manchester, UK, the regional offices in each geography, and the restaurants themselves.

IFB wants to improve business efficiency significantly by providing each employee with a tailored set of tools needed to perform his or her tasks in an easy-to-use and personalized user interface delivered to a wide range of client devices. The resulting system should provide more complete automation and integration of IT services. By delivering a streamlined employee training infrastructure, training costs can be reduced while improving the ability to manage workforce skills. The solution must leverage existing legacy systems, yet minimize the overall complexity of the enterprise architecture.

IFB wants a system that will become the primary point of interaction with resources, such as HR functions, financial systems, supply chain systems, e-learning, and key performance indicators, as well as provide new ways to communicate and work electronically with others.

4.1.2 Core functions

- ▶ Delivery of company news, policies, and reports to employees, with information filtered according to employee role/position.
- ▶ Access to PeopleSoft HR applications and data.
- ▶ Employee schedule system—Allow employees to see their schedules from both browser and kiosks. Employees will also be able to identify scheduling conflicts through this interface.
- ▶ The creation of a task management system—This system will permit the creation of task lists and assignment of tasks to employees, document expectations and task completion goals, and include the tracking and management of tasks.

- ▶ Provide an efficient mechanism to verify inventory and reconcile the central stock management system from the store floor—This needs to integrate with the existing PeopleSoft SCM system.
- ▶ Facilitate efficient employee-to-employee communication, both direct and asynchronous.
- ▶ Delivery analytics of key performance indicators by employee, store, and region to various managers.
- ▶ Provide an education facility to streamline employee training.

4.2 Plan

As in Chapter 3, we will concentrate on the three major tasks that make up the *Execute* phase. To summarize the envisioned goal and vision for IFB, we have captured its business and IT requirements. As always, we must be sure to understand where the customer is today, what it needs to accomplish within the scope of the current project, and its long-term business goals.

4.2.1 Business requirements

The most important features for IFB include:

- ▶ Empower employees and teams to be more productive:
 - Increase employee retention, decrease turnover, and improve HR business processes by automating corporate and store management HR transactions through Employee Self Service
 - Provide every associate with personal access to relevant information about benefits, policies, and procedures and to perform day-to-day HR functions via the Web from the store or home
 - Leverage standard HR applications like PeopleSoft or integrate into legacy applications
 - Create a shared, centralized HR service environment
 - Help them communicate effectively with one another and work together on tasks
- ▶ More streamlined and effective inventory processes:
 - Better tracking of inventory and losses
 - Enabling just-in-time inventory to reduce stocking and spoilage
 - Effective integration with suppliers

- ▶ Optimize task execution and management
 - Set up just-in-time teams appropriate for tasks
 - Provide corporate-wide standardizations through task templates
 - Track and improve task execution

4.2.2 IT requirements

From an IT perspective, the most important considerations include:

- ▶ Ensure security by role and ease of access to different systems
- ▶ Integration with existing back-end systems, including HR, inventory, and supply chain
- ▶ Personalized content delivered by role and device type
- ▶ Manageability of the system through regional IT centers for back-office applications and global applications for teamwork and collaborations

4.3 Execute (develop solution requirements)

We will begin this phase by capturing the system context for the proposed project. This helps establish the project scope and provide a high-level system view to be validated with the customer. It identifies the proposed portal solution (in the center), with the end users on the left and the functional systems on the right.

Note: Building the task management system is also part of this project.

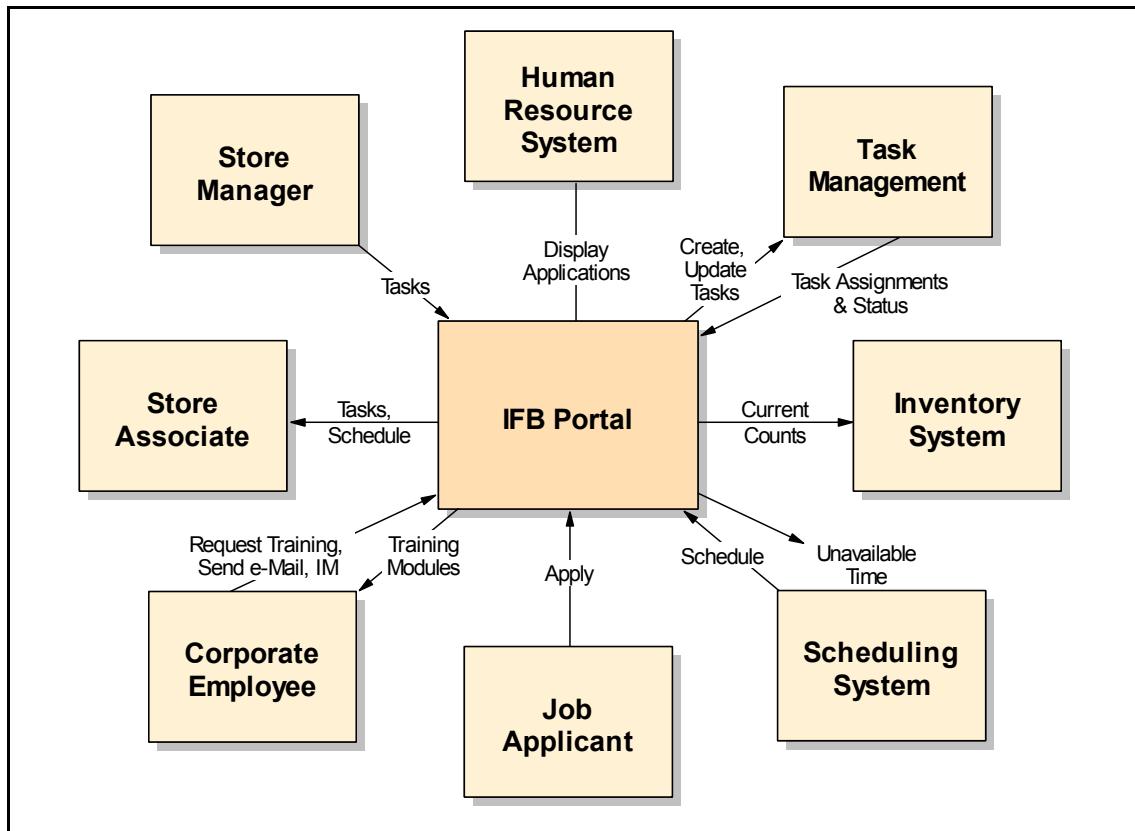


Figure 4-1 IFB system context diagram

Step 1A - List users/existing systems and Functional Requirements

Next we need to document a full list of system users⁵⁴ as well as the functional requirements. Based on the Project Description and the System Context Diagram, we can derive the following list of actors and existing systems and the major functionality needed:

- ▶ Users and existing systems:
 - Store employee
 - Manager
 - Associate
 - Corporate
 - Applicant

⁵⁴ Roles or actors

- Skills database
 - Resumé database
 - Labor scheduling system
 - HR Center of Excellence
 - Task management system
 - Time and attendance system
 - Item replenishment system
 - Point of sale system
 - Back door receiving system
 - Identity management system
- Functionality required:
- Access portal (authenticate and authorize)
 - User management (enroll, deactivate, remove)
 - Fill job vacancy
 - Enter staffing requirements
 - Indicate time away (unavailable)
 - Generate schedules
 - Review schedule
 - Manage compensation and benefits
 - Update personnel records
 - Develop career strategy
 - Assess skill requirements
 - Staff training
 - Manage performance
 - Receive inventory
 - Audit inventory
 - Search for customer service information
 - Create task list
 - Assign tasks
 - Review tasks
 - Track tasks
 - Corporate collaboration (e-mail, e-meeting, IM) Step 1B

From these lists, we derived the initial high-level Use Case Model. This is shown graphically in Figure 4-2 on page 173.

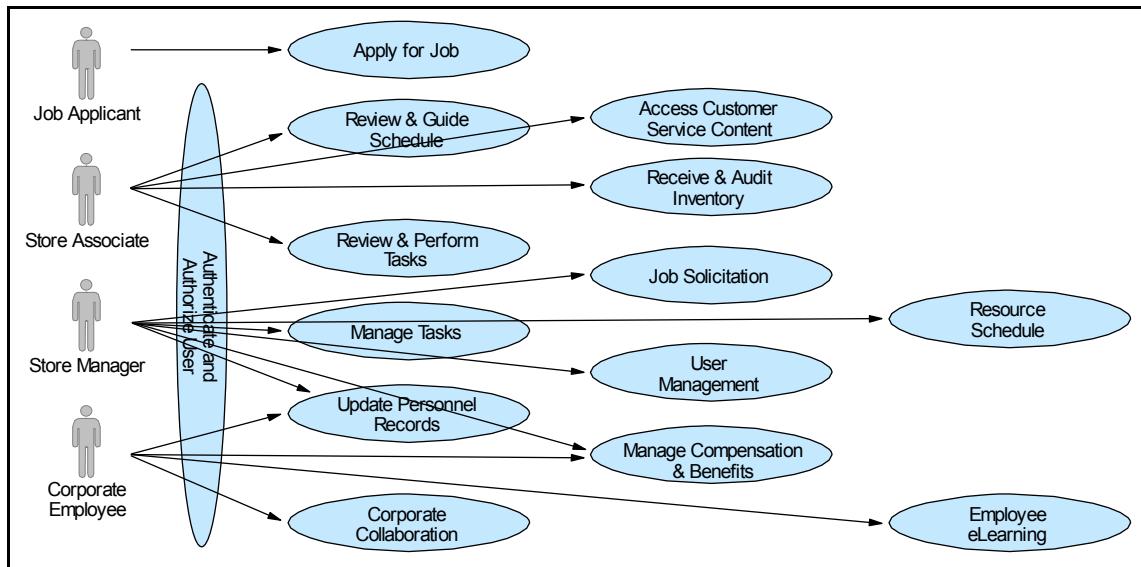


Figure 4-2 IFB Use Case Model

Table 4-1 provides additional detail on the description of these use cases.

Table 4-1 IFB use case descriptions

Authenticate and Authorize User	All internal users authenticate with system, establishing a distinct role for interaction,
Apply for Job	Job applicant submits application
Review and Guide Schedule	Store associate can access current schedule and enter conflicts to affect future schedules
Access Customer Service Content	Store associate can access customer service information to assist a customer
Receive and Audit Inventory	Store associate can perform inventory audit operation and scan in shipments
Review and Perform Tasks	Store associate reviews currently assigned tasks and updates status as tasks are performed
Job Solicitation	Store manager enters staffing requirements and selects applicants to fill vacancies
Manage Tasks	Store manager creates new tasks, assigns staff to tasks and subtasks, and tracks task status
Generate Schedule	Store manager sets inputs to scheduling system

Update Personnel Record	Store manager can modify personnel records of any store employee. Employees can update limited personal data on their own personnel records
User Management	Store manager creates, activates, deactivates, and removes users and role bindings within system
Employee e-Learning	Corporate employees develop career strategy, assess skill requirements, and participate in training
Corporate Collaboration	All employees send and receive e-mail, participate in e-meetings, and participate in synchronous chat
Manage Compensation and Benefits	Store manager can modify employee HR profiles. Employees can access HR services

Step 1C - Inventory large reusable assets

Recall that the next crucial step in the process is to survey the available set of large reusable assets available to you as an architect to leverage. Once again, we will use the Portal composite pattern.

Note: At the conclusion of this exercise we mention that several of the use cases described above that do not match the Portal composite pattern generic use cases are in fact very generic to B2E (ODW) portals. Thus, moving forward you can use this or other ODW solutions as a large reusable asset.

Step 1D - Match solution use cases with generic use cases

Here we categorize those aspects of the current project that match up completely with the reusable asset. In our current task, this involves matching the IFB use cases with the generic use cases of the Portal composite pattern.

Table 4-2 on page 174 show the result of this matching analysis. It should be noted that this initial mapping is somewhat pessimistic. Many of solution use cases can be made to fit into the Portal composite pattern (PCP) under the very broad Access Enterprise Application generic use case. Given that the Task Management functionality and part of the scheduling functionality were also delivered as part of this project, we have deferred including these as a complete match up front.

Table 4-2 Matching the IFB use cases to the generic PCP use cases

Solution Use Cases	Matched PCP Use Cases
Authenticate and Authorize User	<ul style="list-style-type: none"> ▶ Authenticate (Accl) ▶ Authorize (Accl)
Apply for Job	

Solution Use Cases	Matched PCP Use Cases
Review and Guide Schedule	
Access Customer Service Content	<ul style="list-style-type: none"> ▶ Search for Content (IA) ▶ Review Corporate Information (SS)
Receive and Audit Inventory	
Review and Perform Tasks	
Job Solicitation	
Manage Tasks	
Generate Schedule	
Update Personnel Record	Access Enterprise Application (SS)
Manage Compensation and Benefits	Access Enterprise Application (SS)
User Management	
Employee e-learning	
Corporate Collaboration	<ul style="list-style-type: none"> ▶ Synchronous Chat (C) ▶ Participate in e-meeting (C) ▶ Visit TeamRoom (C)

Step 1E - Identify delta use cases

The next step is identifying each of the unmatched (delta) use cases and classifying their associated business and/or integration patterns. Figure 4-3 on page 176 maps the remaining use cases into the primary business patterns, and (where appropriate) to an integration pattern that could potentially be used to integrate this functionality into the baseline solution:

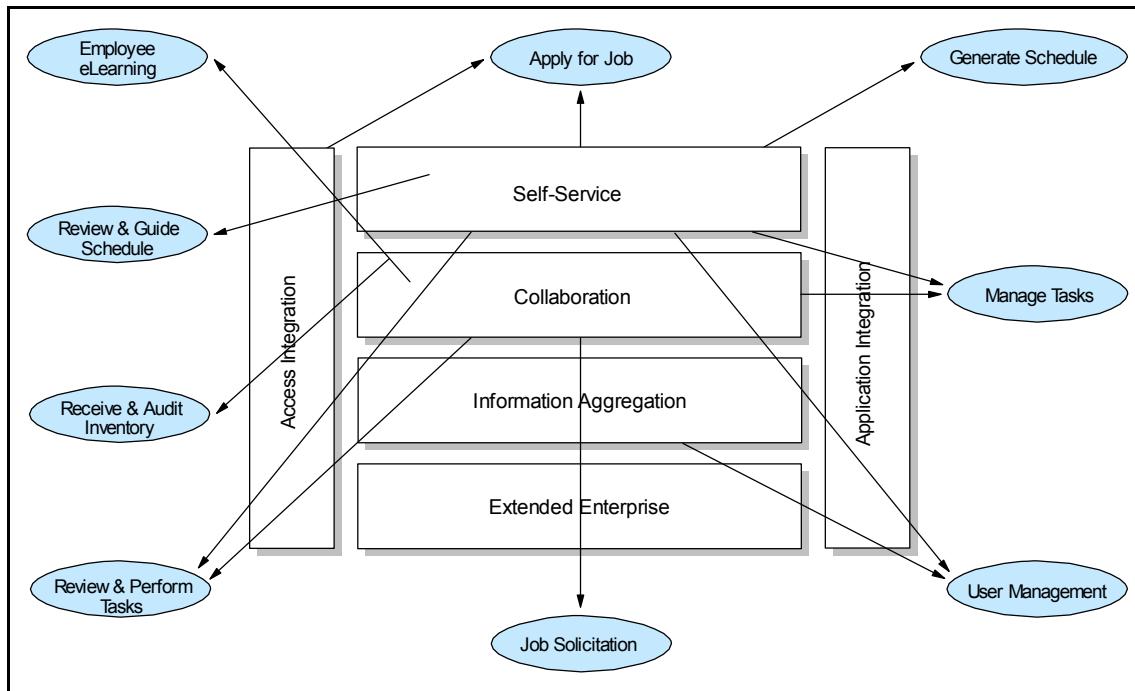


Figure 4-3 Classifying IFB delta use cases

At the business and integration pattern level, none of these delta use cases fall outside the coverage of the Portal composite pattern—in other words, they don't involve either Extended Enterprise or Application Integration patterns.

The final steps in developing the solutions requirements call for documenting the NFRs and performing a walkthrough/validation exercise with the customer.

Step 1F - Identify and document NFRs

The NFRs for the IFB project follow:

Performance

Concurrent accesses from multiple users must be supported.

Availability

Configurable to achieve 24x7, but not required in the first installation—nor is such availability critical

Reliability

The system should respond at least 93% of the time with scheduled maintenance windows allowed.

Response time

- ▶ Access to HR system should be less than two seconds.
- ▶ Access to task management facilities should be less than one second.

Security

- ▶ Public workstations (POS and store kiosks) must not display secure data.
- ▶ All other access types can display only that data appropriate for an authenticated role.
- ▶ Need good single sign-on (SSO) support for back-end systems, particularly the HR system. This is both for user convenience and to ensure proper back-end system access and auditing.

During the walkthrough you will certainly want to validate as many usage scenarios as possible. For example, the lone Security NFR implies that not all system functions will be available to the POS system. You should ensure that meeting this requirement does not exclude the degree of system access expected by the customer.

4.4 Execute (develop the logical design)

Once we have a good handle on the solution requirements we can begin to develop the logical design.

Step 2A - Derive initial AOD

As before, we start by defining the functional subsystems that we will use to seed the initial Architecture Overview Diagram (subsystem). It may be the case that this initial AOD is derived directly from the matched large reusable asset. When such a work product isn't available as part of the matched large reusable asset, you can derive the functional subsystems by grouping the use cases based on related business patterns and common data or processing.

Table 4-3 Functional blocks for IFB AOD

Functional Subsystems	Associated Use Cases
User Management	<ul style="list-style-type: none">▶ User Management (SS)▶ Authenticate and Authorize User (Accl)

Functional Subsystems	Associated Use Cases
Recruitment	<ul style="list-style-type: none"> ▶ Apply for Job (SS) ▶ Job Solicitation (C)
Labor Scheduling	<ul style="list-style-type: none"> ▶ Review and Guide Schedule (SS) ▶ Generate Schedule (SS)
Task Management	<ul style="list-style-type: none"> ▶ Manage Tasks (SS) ▶ Review and Perform Tasks (SS)
Corporate Collaboration	Corporate Collaboration (C)
Access HR	<ul style="list-style-type: none"> ▶ Update Personnel Record (SS) ▶ Manage Compensation and Benefits (SS)
Customer Service	Access Customer Service Content (IA)
Inventory	Receive and Audit Inventory (SS)
Training	Employee e-learning (C)

From here we can draw an initial AOD (subsystem), highlighting the major business functions and their interconnections.

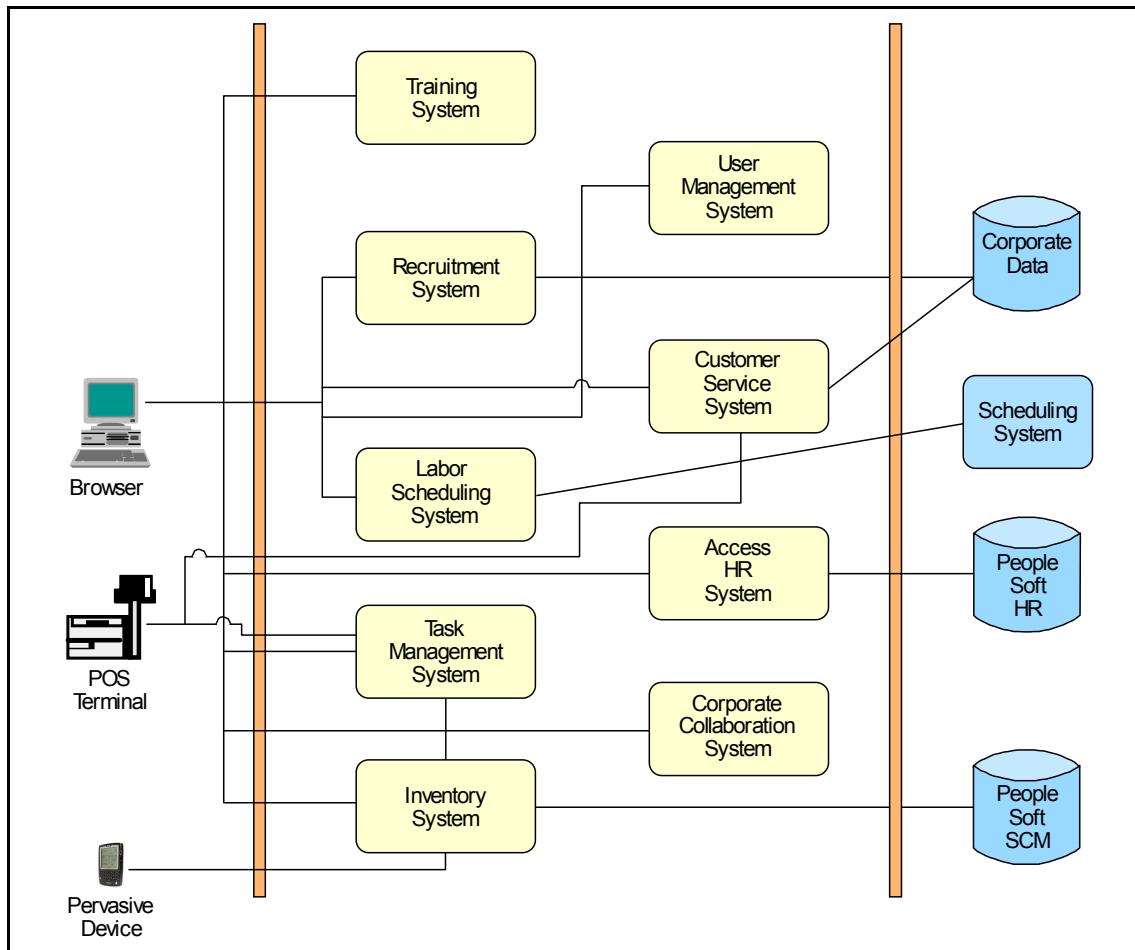


Figure 4-4 IFBs architecture overview diagram (subsystem)

Step 2B - Seed the AOD with large reusable assets

Given the AOD (subsystem), it is once again possible to show the mapping (coverage) of the matched large reusable asset.

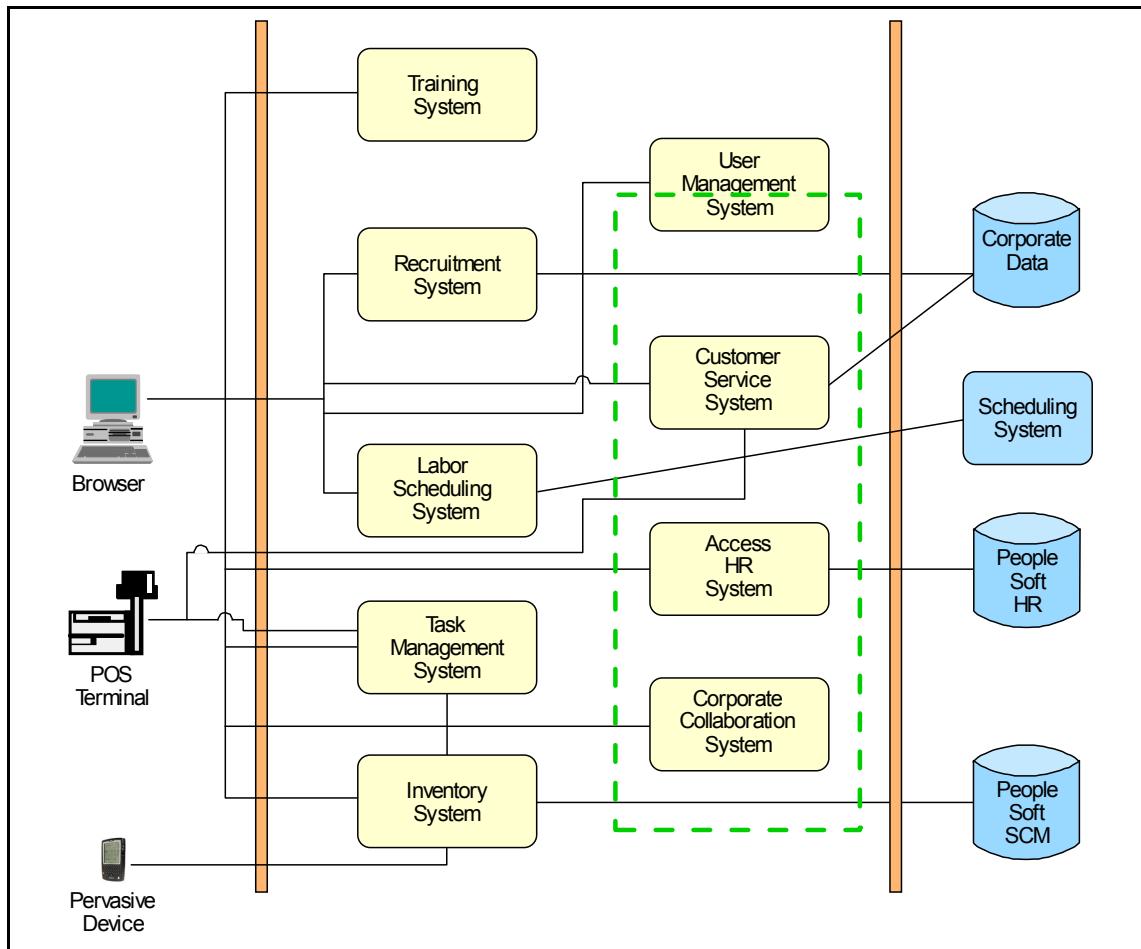


Figure 4-5 AOD (subsystem) highlighting the coverage of the Portal composite pattern

This same information is shown in Table 4-4.

Table 4-4 Subsystems/use cases covered by the PCP (in red)

Functional Subsystems	Associated Use Cases
User Management	<ul style="list-style-type: none">▶ User Management (SS)▶ Authenticate and Authorize User (Accl)
Recruitment	<ul style="list-style-type: none">▶ Apply for Job (SS)▶ Job Solicitation (C)
Labor Scheduling	<ul style="list-style-type: none">▶ Review and Guide Schedule (SS)▶ Generate Schedule (SS)
Task Management	<ul style="list-style-type: none">▶ Manage Tasks (SS)▶ Review and Perform Tasks (SS)
Corporate Collaboration	Corporate Collaboration (C)
Access HR	<ul style="list-style-type: none">▶ Update Personnel Record (SS)▶ Manage Compensation & Benefits (SS)
Customer Service	Access Customer Service Content (IA)
Inventory	Receive and Audit Inventory (SS)
Training	Employee e-learning (C)

Since the matched large reusable asset is the Portal composite pattern, as a final step in validating the coverage we can look at the Application patterns associated with the “covered” functionality to be sure it is consistent with the PCP⁵⁵

⁵⁵ Portal composite pattern

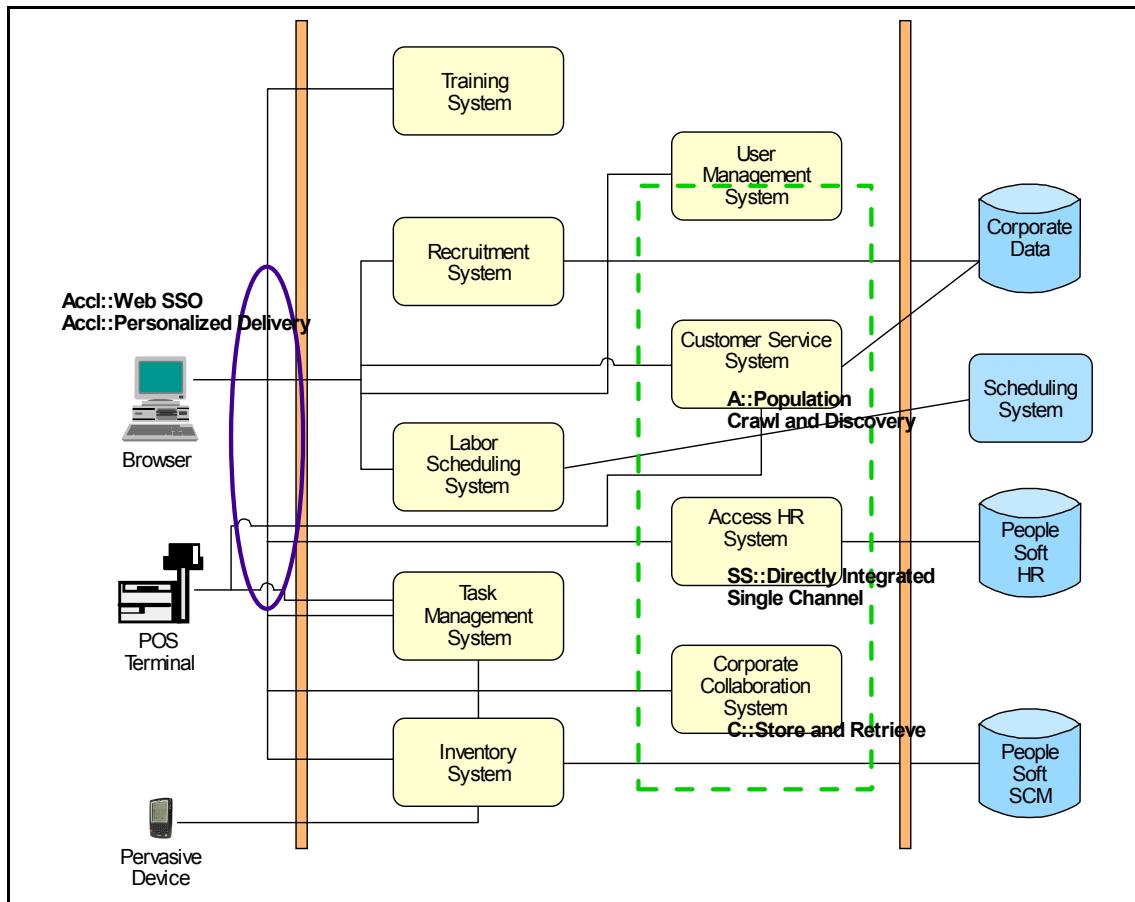


Figure 4-6 Validating Application patterns for function subsystems included within the PCP

Step 2C - Identify and apply patterns for each AOD delta requirement

Next, the delta requirements must be addressed—first identifying the core business and/or integration patterns involved for each requirement, and then identifying the specific Application patterns.

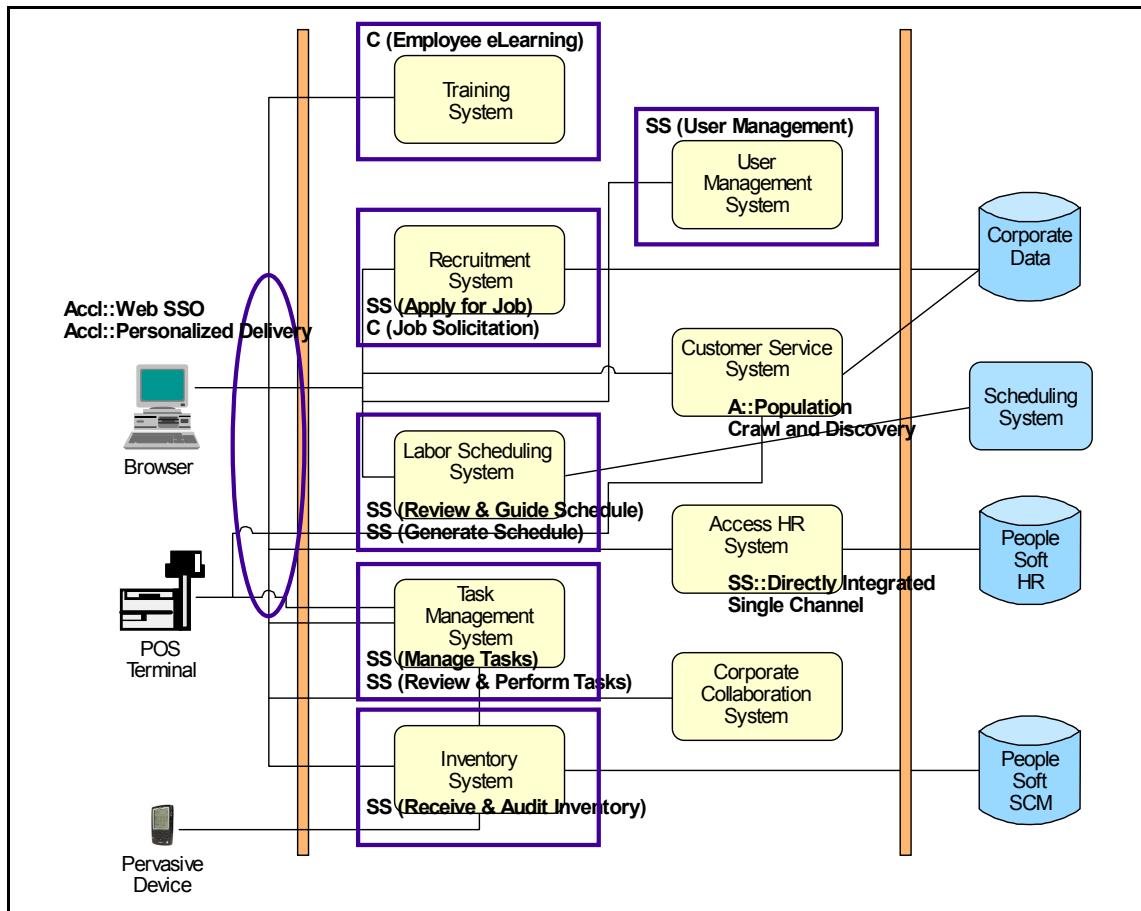


Figure 4-7 Business and Integration patterns for delta requirements

Table 4-5 describes the Application pattern selection for each of the delta requirements.

Table 4-5 Summary of Application patterns for delta Business and Integration patterns

e-Learning System	C::Managed Collaboration—Due to workflow directed collaboration.
Job Solicitation	C::Store and Retrieve—Simple document publishing
Apply for Job	<ul style="list-style-type: none">▶ SS::Directly Integrated Single Channel—Single delivery channel and monolithic application▶ Accl::Pervasive Device Access
Review and Guide Schedule	<ul style="list-style-type: none">▶ SS::Directly Integrated Single Channel—Multiple delivery channels, but can be handled via single presentation server (in other words, Portal)▶ Accl::Pervasive Device Access
Generate Schedule	SS::Directly Integrated Single Channel—Single delivery channel and monolithic application
Manage Tasks	SS::Directly Integrated Single Channel—Single delivery channel and monolithic application
Review and Perform Tasks	<ul style="list-style-type: none">▶ SS::Directly Integrated Single Channel—Multiple delivery channels, but can be handled via single presentation server (in other words, Portal)▶ Accl::Pervasive Device Access
Receive and Audit Inventory	<ul style="list-style-type: none">▶ SS::Router—Multiple delivery channels requiring multiple presentation servers▶ Accl::Pervasive Device Access
User Management	SS::Directly Integrated Single Channel—Single delivery channel and monolithic application

Finally, these Application patterns can be shown in the AOD (Figure 4-8 on page 185).

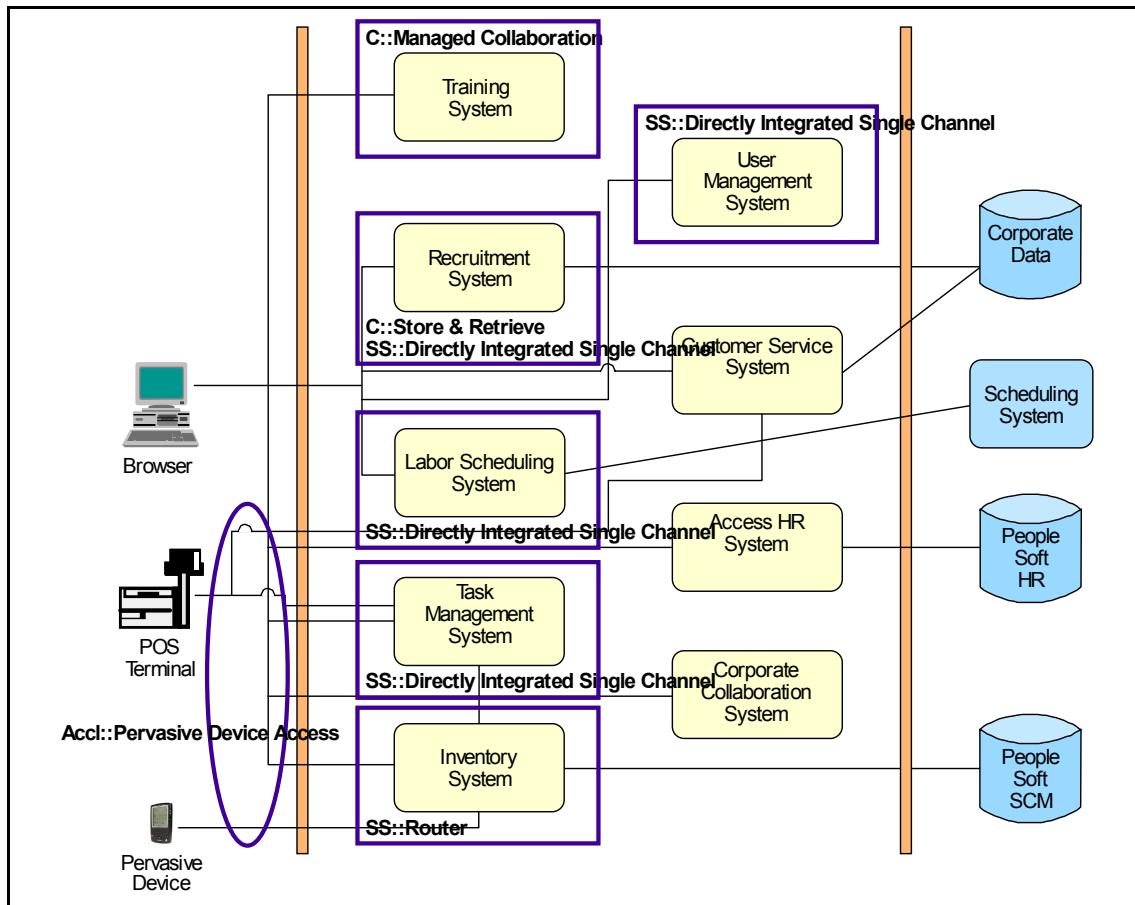


Figure 4-8 Application patterns shown in the AOD

4.4.1 Transition to AOD (logical nodes)

Step 2D - Transition the AOD from subsystem view to logical node view

The next step is to drive down to the runtime pattern level for the solution. Once again, we have seeded the biggest chunk of the AOD (logical nodes) from the corresponding work product of the matched large reusable asset. Figure 4-9 on page 186 shows a slightly customized PCP runtime pattern used to seed the AOD (logical nodes).

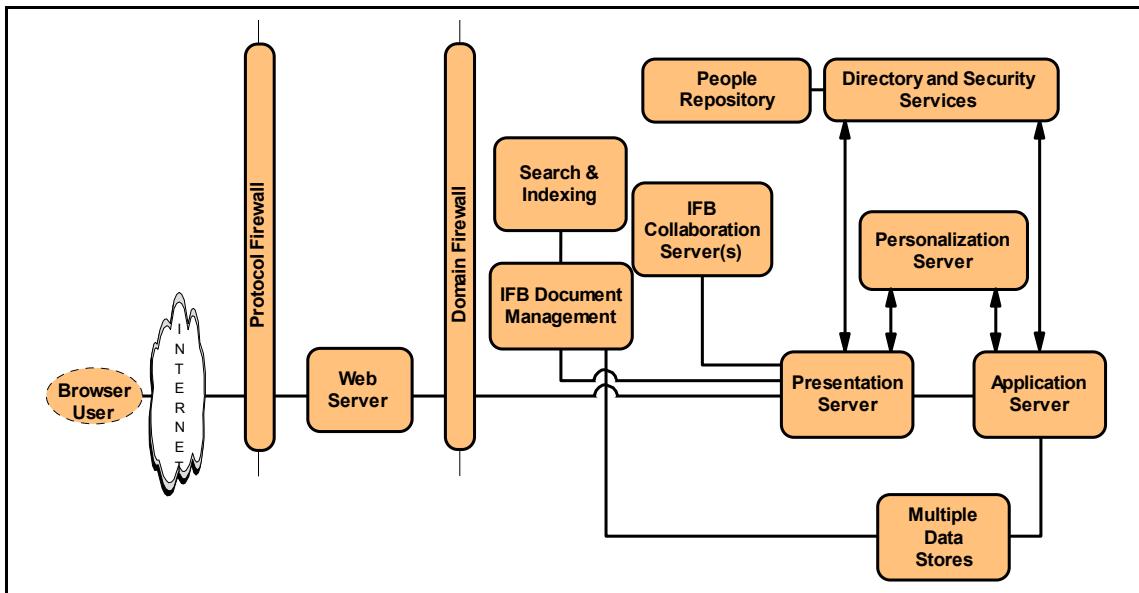


Figure 4-9 AOD (logical nodes) seeded from the customized PCP runtime pattern

Once the initial AOD (logical nodes) is established, each of the delta application patterns can have their individual runtime patterns analyzed to determine how to integrate it within the developing solution. Figure 4-10 on page 187 illustrates adding the wireless gateway node to the AOD and overlaying the Accl::Pervasive Device Access runtime pattern.

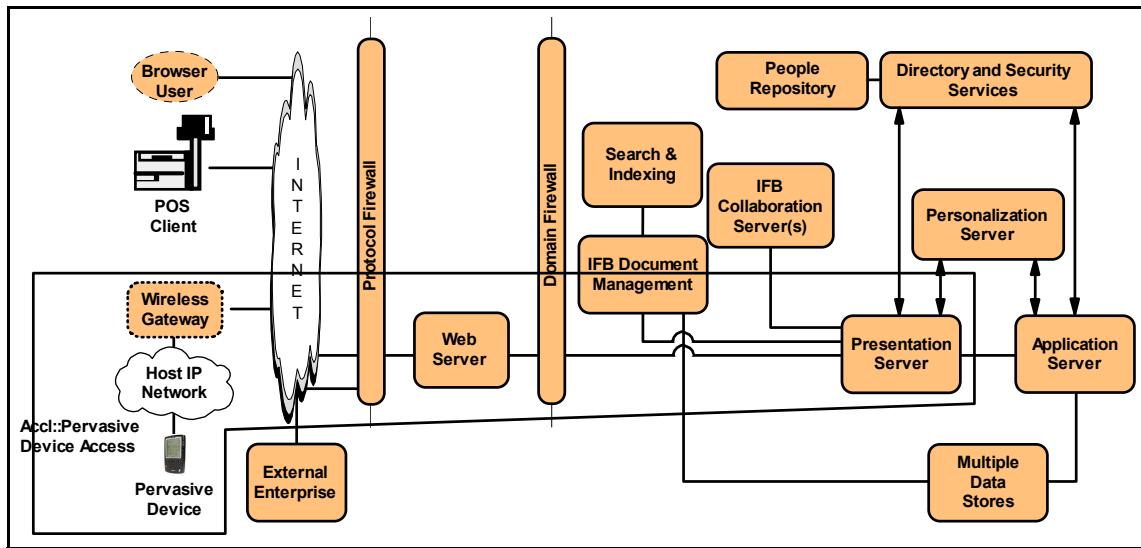


Figure 4-10 Adding Pervasive Device Access support

Figure 4-11 demonstrates adding an integration server to support the runtime for the SS::Router application pattern.

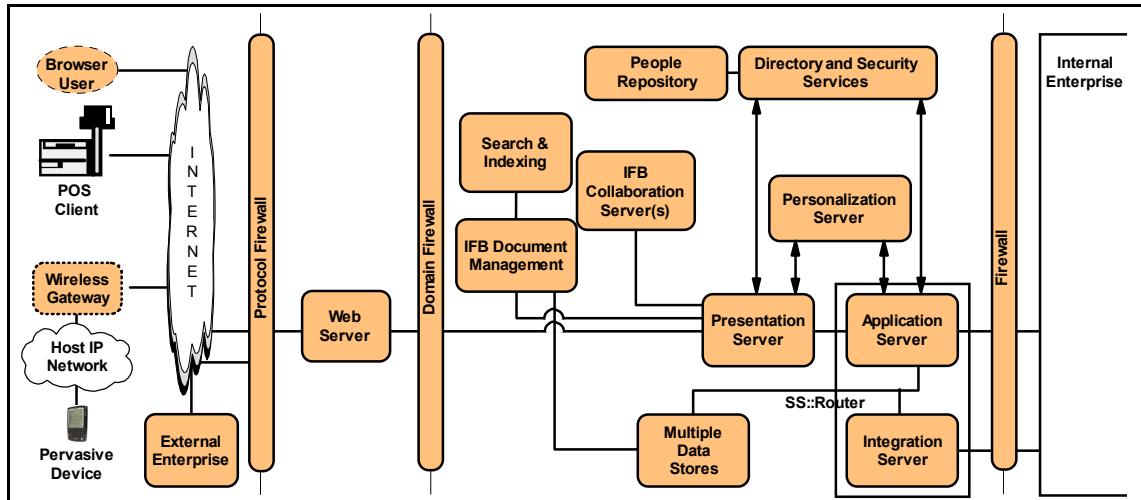


Figure 4-11 Adding integration server support

Figure 4-12 on page 188 represents the complete runtime.

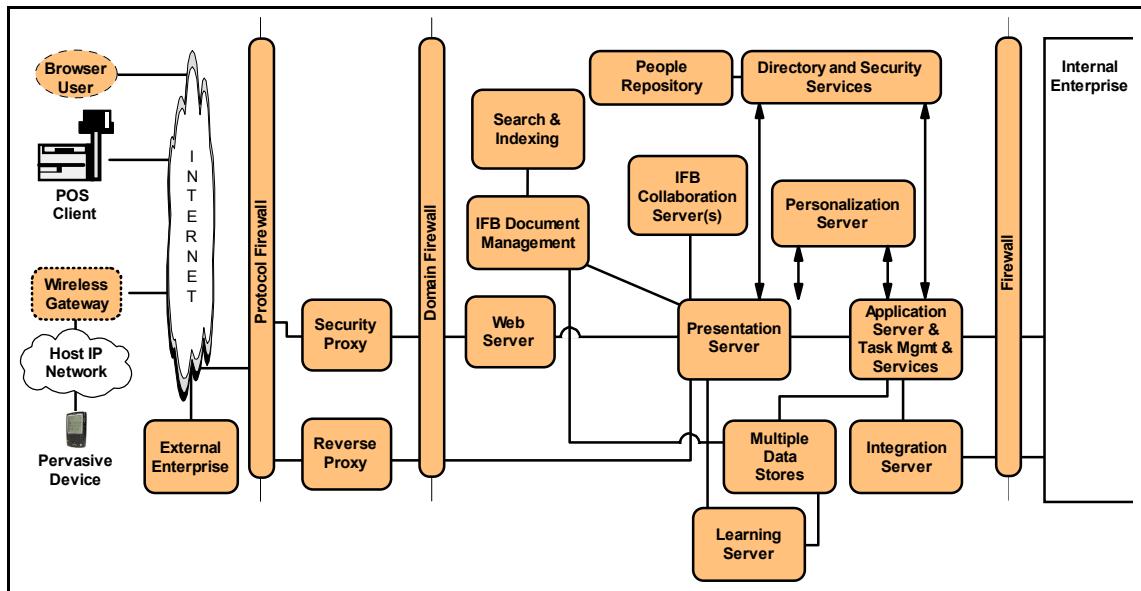


Figure 4-12 Final AOD (logical nodes) for IFB

Table 4-6 depicts the additional decisions made in coming up with the final AOD shown in Figure 4-12.

Table 4-6 Summary of decisions for additional logical nodes to support requirements

Functional or Non-Functional requirement	Architecture Decision (supporting logical nodes))
eLearning System	Supported with Learning Server node
Inventory System (multiple delivery channels)	Supported with Integration Server node
NFR—Security	Moved Web server to internal network and added Security proxy node
Presentation server access to external content	Supported with Reverse proxy node in DMZ

Step 2F - Conduct technical walkthroughs to ensure that functional requirements are met

Here we validate the completeness of the logical solution by walking through each use case. Table 4-7 on page 189 provides the steps for the *Update Personnel Record* use case. The paths associated with each step are highlighted on Figure 4-13 on page 189. Note that this specific walkthrough highlights the fact that there should be direct connections between both the presentation and collaboration servers and the internal enterprise (PeopleSoft HR system).

Table 4-7 Associate update personnel information use case flow

Step	Description	Comments on Diagram
1	An Associate enters People Management Solutions via the Dynamic Workplaces™ Portal	Standard login and navigation from a browser client to the ODW.
2	The Associate reviews his or her personnel information	The presentation server utilizes a connector to the HR system providing the key data items (appropriately filtered for a specific Associate) on the ODW desktop.
3	The Associate chooses to update his or her address	
4	A notification is posted to the Operational Manager and the HR system	The same connector interface permits updates to specific fields, again based on role and specific user (standard notification behavior of HR system for data field updates).
5	The HR system posts an update notification to the Associate	Standard notification behavior of HR system for data field updates

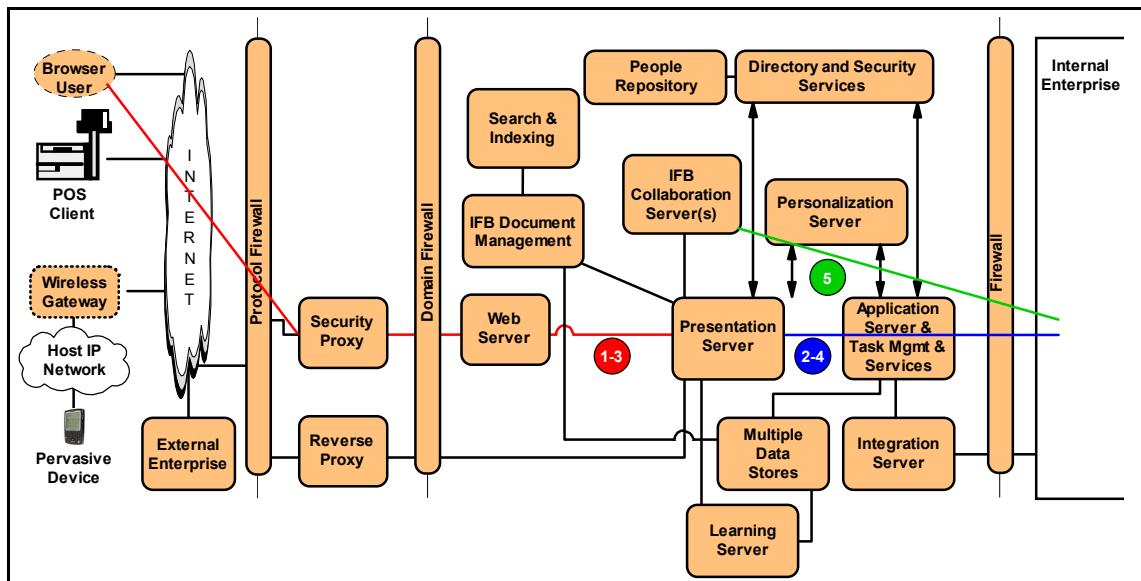


Figure 4-13 Walkthrough for the Update Personnel Records use case

4.5 Execute (develop the physical design)

Step 3A-C⁵⁶

The Component Relationship Diagram for IFB is shown in Figure 4-14 on page 190.

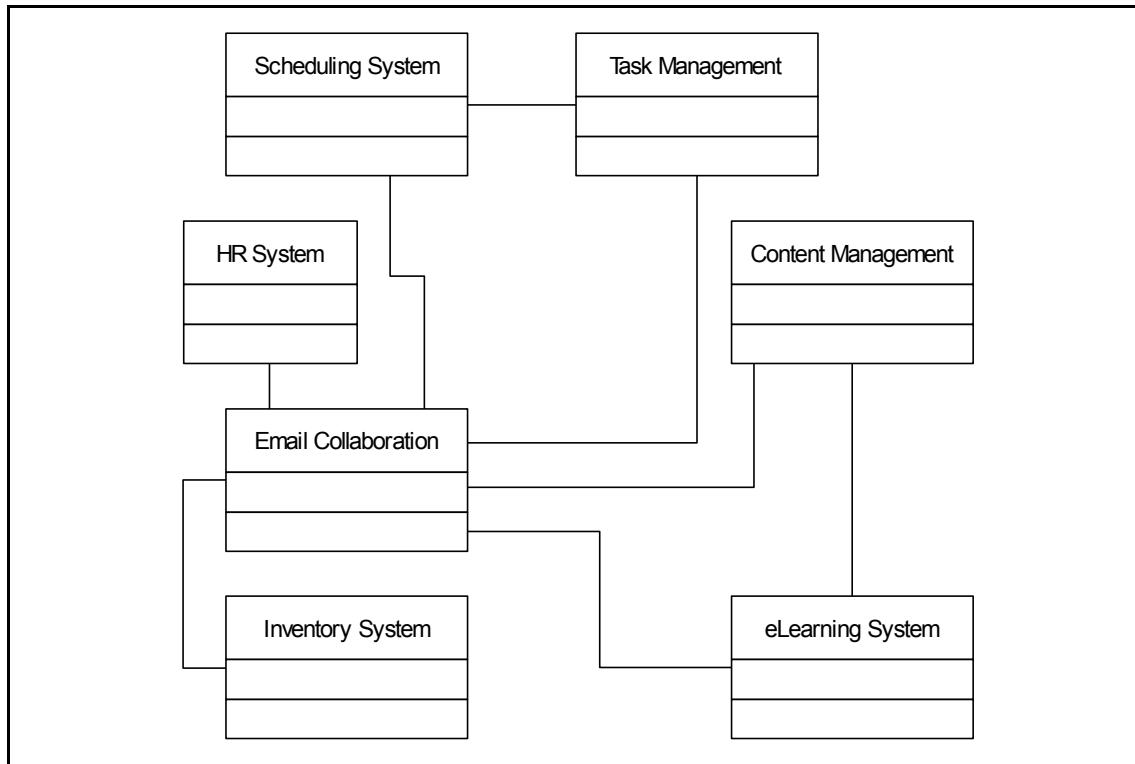


Figure 4-14 IFB component relationship diagram

Component interaction diagrams are useful to understand the details of the interaction between components in supporting use cases.

⁵⁶ Step 3A - Build the Component Model

Step 3B - Apply design guidelines (application level) from matched large reusable assets solution:development methodology:

Step 3C - Conduct technical walkthrough of component interaction

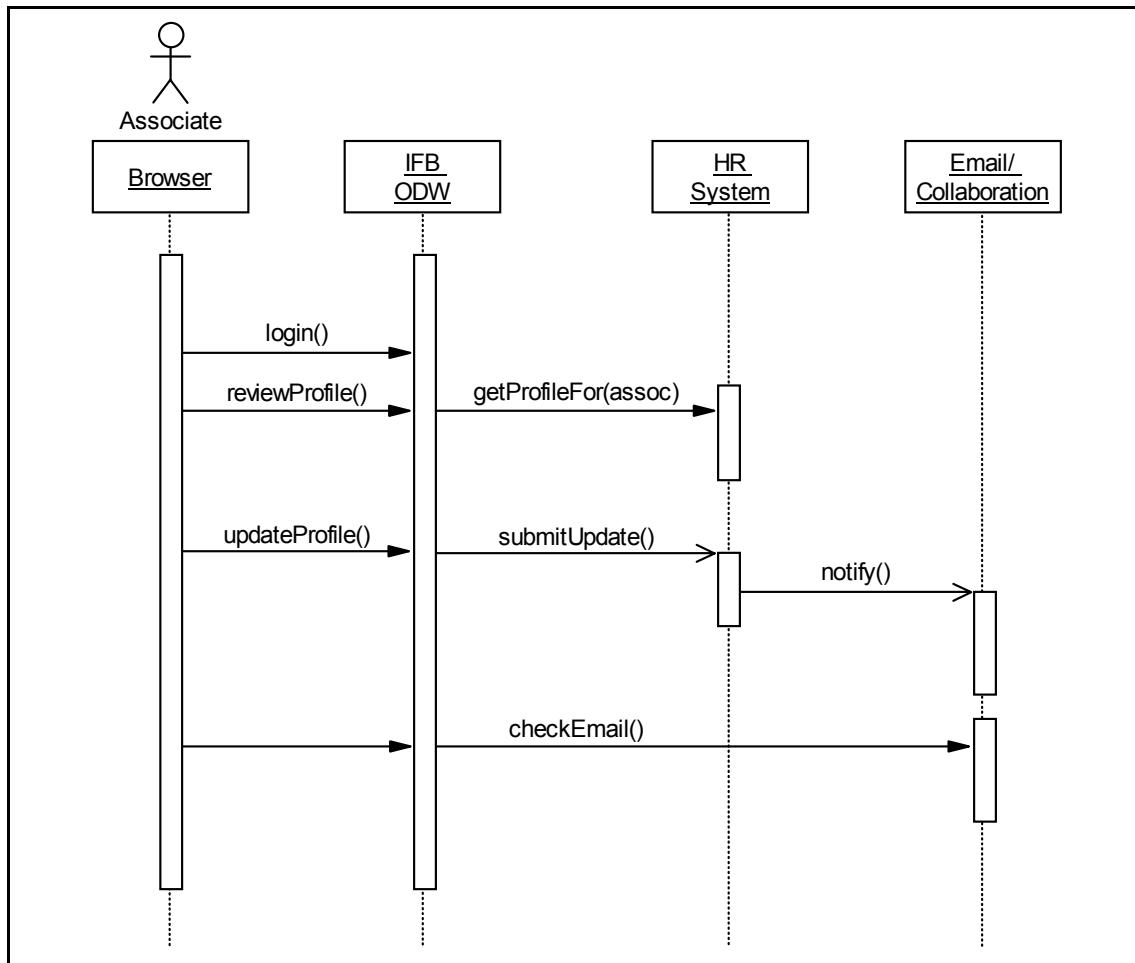


Figure 4-15 IFB component interaction diagram—Update personnel information

Figure 4-16 on page 192 identifies the software services associated with each of the logical nodes. This is in preparation for mapping to software products that provide those services as well as mapping software components to physical nodes in an operational model.

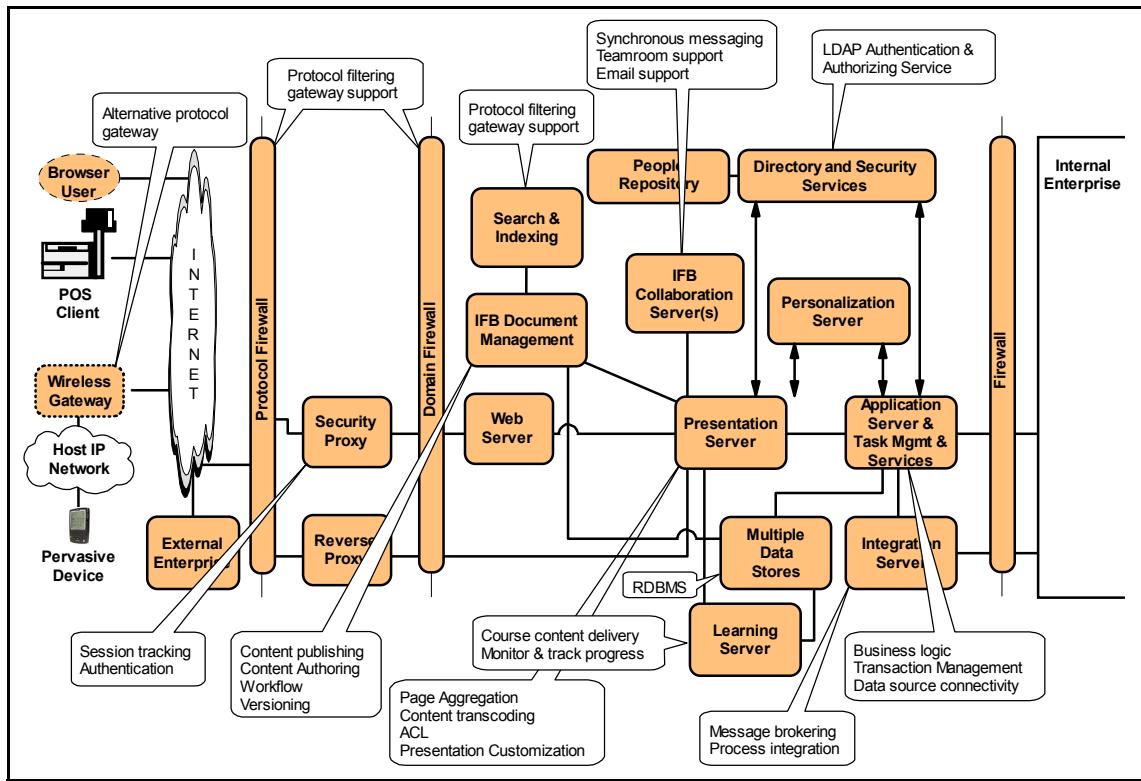


Figure 4-16 IFB AOD with identified software services

Step 3D - Develop the Operational Model

Figure 4-16 show an initial operational model with the assignment of software services to software products mapped to physical nodes. One important operational model decision not visible in Figure 4-16 is the use of the WebSphere Portal Application Integration framework to provide direct integration with the PeopleSoft back-end systems from the portal server.

Step 3E - Seed the Operational Model

Figure 4-17 on page 193 shows product mappings applied to the Operational Model.

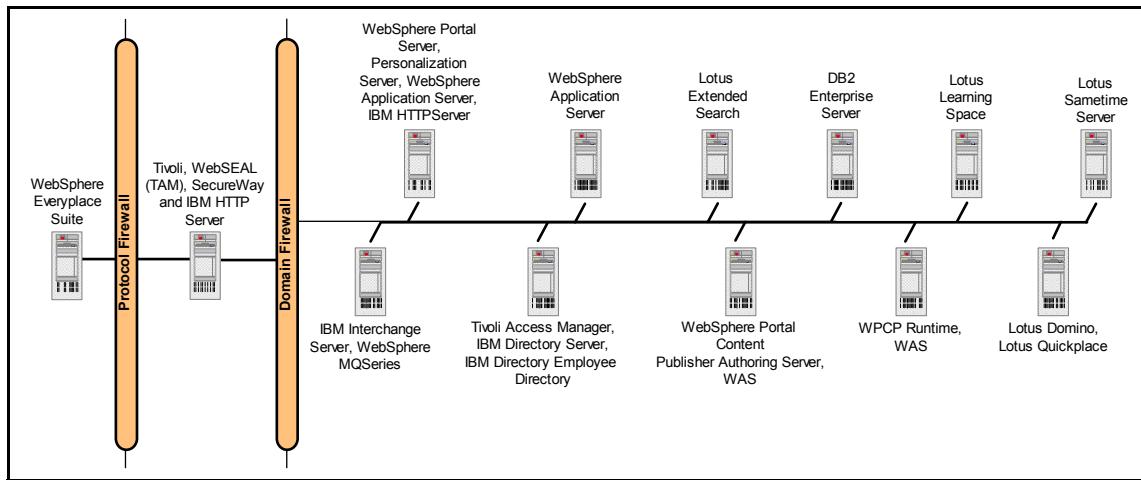


Figure 4-17 IFB Operational Model

Step 3F - Address operational NFRs in the Operational Model

None of the NFRs require any specific modification of the operational model. The response-time NFRs may affect machine sizing. With the exception of the access restriction to both POS and kiosk devices, the security NFRs are covered by Tivoli Access Manager. Access management is controlled through specific roles established for each device type, with access rights architected to enforce security requirements.

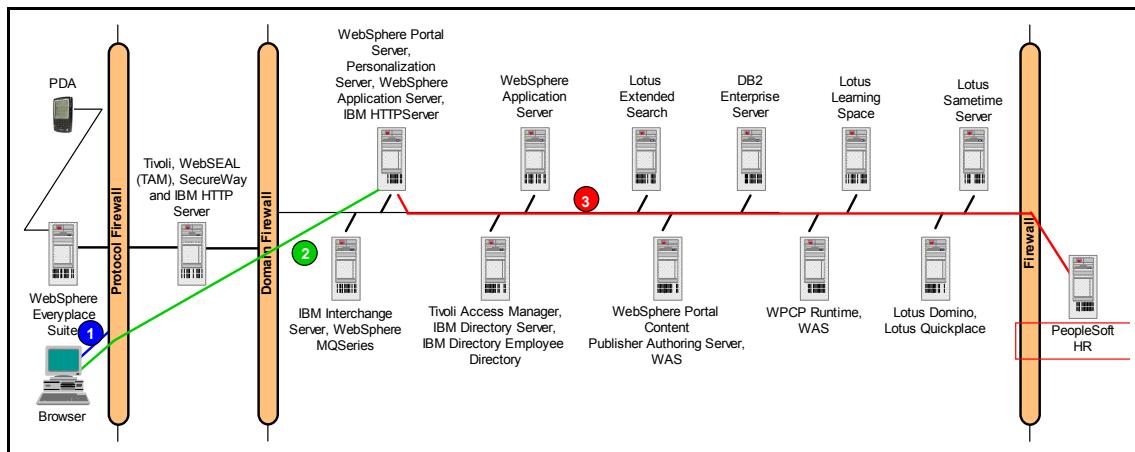


Figure 4-18 SSO walkthrough using Operational Model

Table 4-8 Description of SSO walkthrough

Step 1	Associate gets prompted for portal credentials from WebSEAL
Step 2	Credentials are provided to portal, and the Associate navigates to the HR page to request information about his/her current personnel profile
Step 3	PeopleSoft portlet (WebSphere Portal Application Integration) accesses the Portal Credential Vault to retrieve the PeopleSoft credential for the Associate. This credential is presented to the PeopleSoft application, along with the request for the personnel profile data.

Step 3G - Conduct technical walkthroughs to ensure that NFRs are met

Figure 4-18 on page 194 and Table 4-8 on page 194 together describe a walkthrough of the Operational Model, validating coverage of the Single Sign-On NFR.

4.6 Reflection on ODW

Throughout this chapter we have referred to a broad category of portal solutions that can be characterized as *On Demand Workplaces*. Mining the assets from solution architectures such as IFB and deriving generic “large reusable assets” can be a valuable activity. Here we describe a first cut at an ODW composite pattern. We highlight this asset by showing generic use cases and mandatory business, integration, and application patterns, and provide a composite runtime pattern.

4.6.1 ODW generic use cases

The following generic use cases have been identified for an ODW composite pattern. Here we only show those use cases that are in addition to the generic use cases associated with the Portal composite pattern:

- ▶ View Syndicated Content
- ▶ Access from PDA
- ▶ Access eHR System
- ▶ View Corporate Information
- ▶ Manage Customer Accounts
- ▶ Manage Supplier Accounts
- ▶ Do Online Training
- ▶ Access Help Desk
- ▶ View Analytic Data

In the IFB case study considered in this chapter, several of these generic use cases appear. In fact, had we had a reference (ODW) architecture, the following additional use case matches would appear:

Table 4-9 Matching case study use cases to ODW generic use cases

Solution Use Case	ODW generic Use Case
Access Customer Service Content	Access Help Desk
Update Personnel Record	Access eHR System
Employee e-learning	Do Online Training
Receive and Audit Inventory	Access from PDA (partial)

From the generic use cases, we see the following additional mandatory application patterns:

- ▶ Access Integration::Pervasive Device Access (required by Access from PDA)

- ▶ Access Integration::Extended Single Sign-On (typically required by Access eHR)
- ▶ Collaboration::Directed Collaboration (several ODW collaboration services)

Many specific ODW solutions will incorporate Application Integration patterns, but in this initial cut we are keeping it as an optional integration pattern for the ODW composite pattern. This is also shown in the ODW runtime pattern with an optional integration server. The other optional configuration in this pattern is the presence of a security proxy in the DMZ (typical), with the Web server in the trusted network.⁵⁷ The resulting runtime is very similar to the IFB runtime pattern (Figure 4-19).

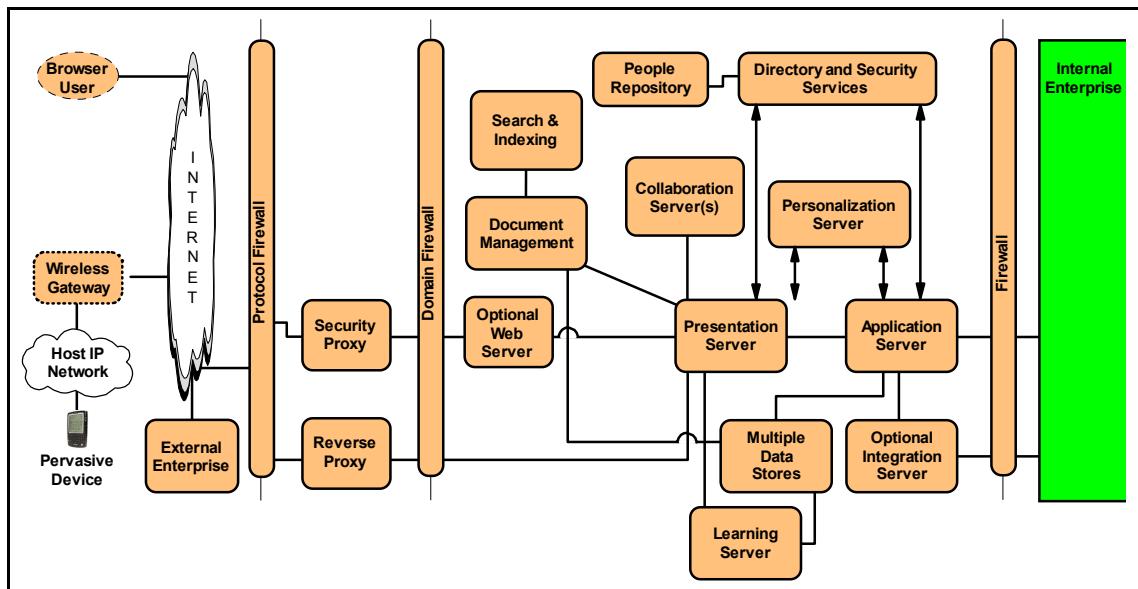


Figure 4-19 ODW composite runtime pattern

There is ongoing work within IBM to formalize this ODW composite pattern and further establish additional work products as well as design guidelines/best practices to support it. Mining a composite pattern from a solution has only been presented informally in this publication. It should be noted that this “leap” was based on extensive work not shown here, where a recurring set of functional requirements and use cases appeared in multiple On Demand Workplace solutions developed within IBM. Pattern mining in general requires observation of such patterns appearing in multiple solutions.

⁵⁷ Alternatively, the Web server could move to the DMZ with or without the security proxy.

The existence of a formal ODW “solution asset” could have been used as the large reusable asset for the IFB solution development process. In such a scenario, a larger portion of our solution could have been seeded from this (ODW) large reusable asset.

Extending portal solutions with IBM Business Partners

Building complex e-business on demand solutions will not be possible without extensive cooperation with hardware and software industry leaders, IBM

Business Partners, and customers. It is common practice for IBM to work with its large set of business partners to plan, design, deploy, and maintain these complex solutions for customers. The key to making this collaboration work effectively is to ensure that the same methodologies, best practices, and reusable assets are shared and used by all the communities within the development ecosystem.

Part 2 of the redbook showcases how business partner Galaxia extended the approach to architecting portal solutions articulated in Part 1 to incorporate its solutions and offerings in conjunction with IBM. This chapter provides a blueprint for other business partners who want to include their solutions effectively with IBM and also to leverage reusable assets and best practices in building solutions.



Galaxia and the On Demand Workplace

This section showcases Galaxia's ePCM,⁵⁸ which enables an On Demand Workplace for the Automotive, Aerospace, and Transportation industries. We first provide an overview of the challenges faced by these industries and a discussion of the key processes, and then proceed with an explanation of how the solution architecture has been conceived. We study the business and IT drivers to crystallize a general set of requirements and map it to "the best practices" (patterns). Once the patterns are identified, we map the appropriate products necessary to build the solution. Finally, we illustrate a scenario that uses this solution and provide more detail on the features and benefits of ePCM.

⁵⁸ Enterprise-wide Process Change Management

5.1 Galaxia solutions for the industrial sector

Galaxia is a solutions provider with proven expertise in industrial service sectors, including aerospace, automotive, and transportation. Our work with IBM has helped us to be at the cutting edge of technology in the areas in which we operate and to think of ways in which the processes of our customers can be improved, specifically through the use of portal-based solutions.

5.1.1 Galaxia and ePCM

Galaxia's core competencies include Business Process Modeling and Integration for companies in the Aerospace, Automotive, and Transportation sectors.

ePCM is a set of comprehensive process-centric solutions for an "Enterprise Workflow" that allows you to better manage and monitor business processes. It provides companies the benefit of having a collaborative enterprise-wide application without having to expend the long implementation time, high costs, and inflexibility of other BPA systems.⁵⁹

Galaxia ePCM solutions are Web-based and allow for easy collaboration between various locations in the company and suppliers, thus extending the enterprise. Galaxia can easily map and model enterprise processes for BPA using ePCM, with templates for various business processes. ePCM - Portal version takes advantage of the portal infrastructure to provide additional "out-of-the-box" functionality and advantages as shown later in this chapter.

5.1.2 Industry overview

Automotive, aerospace, and transportation

Automotive companies are the pioneers of modern manufacturing techniques. Ford and others championed the assembly-line concept to enable mass production. In the past, customers didn't have much choice. Market growth was unlimited and customers were easily satisfied.

Today, manufacturers compete globally, and the trend toward make-to-order manufacturing⁶⁰ has put new demands on manufacturers and dealers. Customers want *what* they want *when* they want it. Today's market is highly fragmented, and manufacturers have to deal with myriads of product configurations and models. In response to new demands, new models and designs are added, thus increasing the complexity of the designing and manufacturing processes.

⁵⁹ Business Process Automation

⁶⁰ Filling individual customer requirements rather than make-to-stock

With the onslaught of globalization and intense competition, the manufacturing process has become distributed and follows economies of scale. Parts and components are designed and built at multiple locations by the same manufacturer. In order to remain competitive, manufacturers are now outsourcing both the design and assembly of cars. Digital manufacturing and designing techniques along with efficiencies that e-business can offer will become increasingly important at all points along the value chain.

Competitive advantage depends on speed—in product development, assembly, and delivery to dealers and customers. Manufacturers must closely manage manufacturing processes to ensure that the correct parts and components are delivered when and where they are needed. Collaboration among all the industry players (suppliers, OEMs, dealers, and customers) is thus essential.

In the Automotive, Aerospace, and Transportation industries, major challenges include:

- ▶ Changing customer needs
- ▶ New competition on the horizon
- ▶ Unviable cost structure
- ▶ Dwindling revenues and increasing costs
- ▶ Excess capacity
- ▶ Government regulations
- ▶ Ineffective processes/workflows and decision support systems

It is imperative that these industries identify and improve some of their inherent inefficiencies in order to respond effectively and sustain a viable operation. Most companies have processes and methodology to manufacture a vehicle. The manufacturing process is no longer isolated from the external world. Hence, the competitive edge is how well these processes are defined and fine-tuned to respond to changes externally and internally.

The manufacturing process should interact with other enterprise processes to provide seamless integration and collaboration of different internal and external organizations. When we look at some of the key processes in vehicle development, we see that it requires a concerted effort from all involved as well as an efficient environment to manage the development cycle.

5.1.3 Vehicle development process

Let's look at a typical vehicle development process and outline various phases required to get a better understanding.

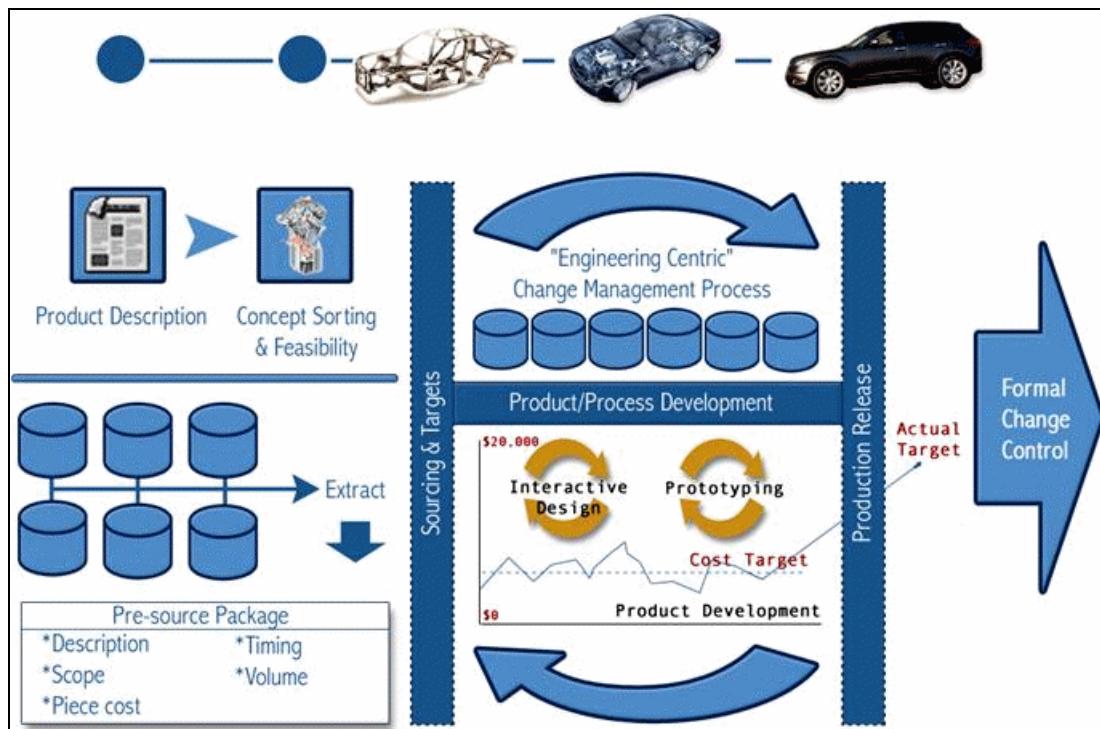


Figure 5-1 Vehicle development process

Product description

This activity defines targets for vehicle design. The characteristics of the design target are defined based on the marketing plan of the company and the market segment to which the vehicle appeals, defining targets such as performance, styling, and cost at a high level.

Concept sorting and feasibility analysis

The next step is to break down the high level first into sub-systems and then down to the individual components. This effort eventually leads us to a tree-like product structure called *BOM*.⁶¹ The targets for each component are defined in such a way as to achieve the overall design goals.

Development

Once the targets have been defined, designers using CAD/CAM systems design them. These tools also do simulation to virtually test individual designs so that they meet the targets. The development activity is no longer a single location or in-house activity. Companies are outsourcing some design and development activities to key partners and suppliers. The development activity involves individual component-level testing of each component and system-level testing to ensure that the results are within acceptable limits. Digital mock-ups of the assembly and individual component testing often leads to rework and redesign. Hence, design and development is a highly iterative process requiring constant collaboration with those involved in this process.

Prototyping

Once the initial design is validated and prototype parts become available, actual testing of individual parts and assembly will confirm the virtual test results. The physical testing phase is also an iterative process and may lead to additional changes to conform to the targets defined. Design costs can be quite high, since some degree of commitment to tooling and manufacturing can occur at this stage. Thus, identifying probable design snags at an earlier stage can significantly reduce product development costs. Separate testing teams for validating the Production Release Design usually do physical product testing and validation. Testing and validation requires intense collaboration with designers and the various testing teams. Adding to this complexity is outsourcing design of certain parts to key partners and suppliers.

Production release

The design is eventually validated and ready to be released for production. Commitments to tooling and manufacturing are made at this stage and production commences, meaning that any further design changes can have severe time-to-market and final cost repercussions. Since considerable cost is locked in at the design stage of Product Development, the key is to keep *actual costs* close to *target costs*.

⁶¹ Bill of Materials

Change management

The company's PCM⁶² process is thus critical to the design and development phase. Design is usually an iterative process validated by different people. Design changes and even redesign are not uncommon. They can be time-consuming and commonly escalate schedules. Therefore, rigorous monitoring and tight controls are required to govern the development cycle. The data collected during a design change is quite valuable and may even prove to constitute business intelligence. Such information helps the design team share its experiences with other teams and avoid future rework. Most companies have some form of processes in place to deal with such change requests, but many of these are paper-based, have long cycle times, and are not rigorously tracked, controlled, or integrated with other processes.

Following are brief descriptions of some of the more common PCM processes:

RFC (Request for Change)

The Request for Change is a key process that addresses engineering and non-engineering related change management issues. The RFC may or may not result in an ECR.

⁶² Production Change Management

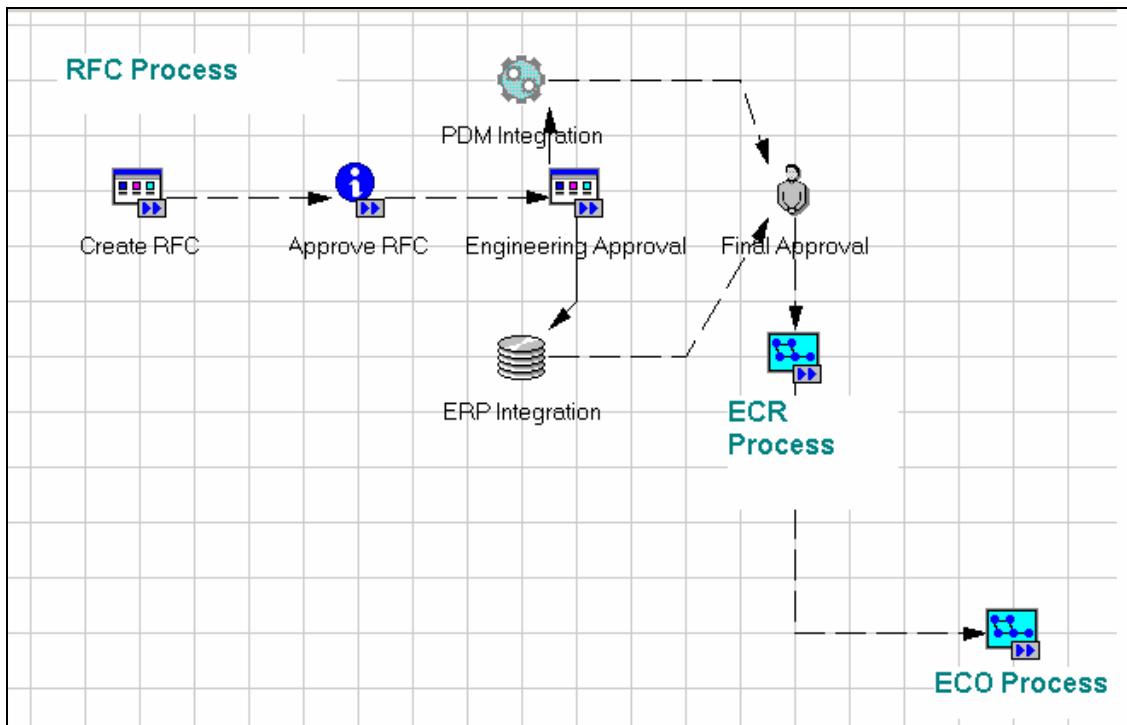


Figure 5-2 Change management processes

ECR (Engineering Change Request)

This is either a subset of the RFC or a completely separate process that addresses issues regarding a particular engineering change. The process usually begins with the creation of an ECR, and as it advances through the process integrates data from other applications to enable better decision-making. An approved ECR usually results in an ECO.

ECO (Engineering Change Order)

An ECO is the result of an ECR approval. This process involves sharing the new design with the concerned departments and then finalizing the change.

Conclusion

The vehicle development process aims at designing and producing quality products within a short timeline and with minimal cost. Achieving this somewhat utopian goal requires that the business be *responsive*, *variable*, *focused*, and *resilient*. To attain these *on demand* characteristics, companies must institute an operational environment that is *integrated*, *open*, *virtualized*, and *autonomic*.⁶³

Users spanning various departments within and outside the enterprise need secure access to information and tools that support decision-making in order to work effectively and collaborate. Since information exists in disparate systems, users must employ multiple systems to finish a logical unit of work.⁶⁴ They need simple tools that hide the complexity of these disparate systems and at the same time are flexible enough to allow them to personalize content—an operational environment/workplace that is well-integrated, flexible enough both to utilize existing applications and to add new ones, and that conforms to open as well as defined standards.

Investment in existing infrastructure must be harnessed using a business process integration framework, thus helping companies manage various computing resources as a single shared virtual computer. To conform to an *on demand* environment, companies must use a framework that is open and that conforms to defined standards, thereby promoting interoperability and future system enhancements. Such an environment/workplace will share computing resources that already exist and can be easily managed using a business process integration framework.

5.1.4 Key business drivers

Some of the major challenges posed to the Automotive, Aerospace, and Transportation industries are:

Reduce time to market

With changing customer needs and growing competition, companies have to find ways to shrink “time to market.” There is pressure to reduce the cycle time all the way from the product development stage to market. Since cost for a product is usually locked in at the concept or design stage, it is imperative to “get it right” at the design stage and thereby avoid design slack time. Design being a collaborative process involving the design team, partners, suppliers, and customers, this requires a robust collaborative platform in order to respond efficiently to design change requests.

⁶³ See 2.3.1, “What is an e-business on demand?” on page 58 and 2.3.2, “Building the on demand operating environment” on page 59.

⁶⁴ See Chapter 2, “Portal solutions” on page 39.

Lower costs through operational efficiency

Increased competition and dwindling orders have led companies to focus on improving operational efficiencies, harnessing more mileage out of existing infrastructure, and eliminating non-essential tasks in order to better focus on the core business.

Outsource and collaborate throughout the extended enterprise

Company facilities may be in multiple locations, with some of the core/non-core business activities outsourced to reduce cost and improve efficiency. Alliances with partners in other locations (and perhaps even in different countries) require collaboration and a transparent business process.

Improve global process visibility

Most businesses have some kind of process mapping but are predominantly managed with few if any controls. Many of these systems rely on paper-based activities leading to increased cycle times and with insufficient audit trails constituting lost knowledge.

Especially in the automotive industry where product development is the key focus, a complex process like engineering change cannot be managed manually or incompletely done within a system having limited visibility. Even when engineering or product data is managed through PDM systems, it may still lack the ability to manage the change effectively or to share data at an enterprise level. The complexity increases if there are multiple PDM systems in different physical locations.

Processes may not be clearly defined, leading to ad hoc management for every product change request. Impediments in some cases involve the inability to decide which system will federate the process.

Eliminate slack time in business process by being process/workforce-centric

Product development involves designing, multiple levels of testing, and validation spanning different departments. This is an iterative process that follows certain business rules, government regulations, etc. To shorten product development cycle time, users must have the proper visibility to make early decisions. This includes a unified visibility of product information, specifications, and other enterprise-level data.

For example, managers or design team leaders need detailed information about the design from other enterprise-wide applications in order to approve a design or design change. Unavailability of pertinent information at the right time and place may well delay the process and increase cycle time.

Tasks done by each individual is an activity belonging to a *process*. Each task deals with decisions based on information residing in disparate applications or with other users. There is a strong need for an end-to-end workplace so that businesses can optimize every employee's effectiveness.

Eliminate paper-based communication

Engineering Change Requests are usually paper-based. This can lead to pitfalls such as missing information, ineffective capture of the change request, or long cycle times resulting from paper-based approvals.

5.1.5 Key infrastructure and IT drivers

Companies have invested heavily in recent decades on disparate IT technologies and systems catering individually to specific needs.

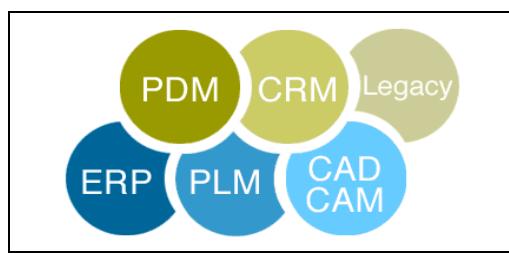


Figure 5-3 Typical IT infrastructure

In the engineering realm, some of these systems include:

CAD/CAM systems⁶⁵

These products are used to design, analyze, and visualize product parts and assembly.

PDM systems⁶⁶

These systems manage design data and product configurations, predominantly engineering data warehouses and repositories that manage product structures and configurations in order to improve Engineering Department productivity. Data warehouses have information *about information* (also called *metadata*). The actual engineering drawing or document is stored in the system where it was originally created. PDM products have engineering-related workflows to manage engineering changes, but these are usually limited to the Engineering Department and do not scale to enterprise-wide workflows.

⁶⁵ Computer Aided Design/Computer Aided Manufacturing

⁶⁶ Product Data Management

PLM systems⁶⁷

These systems aim toward managing products from concept to retirement. They also manage concurrent design and engineering efforts.

ERP systems⁶⁸

These systems manage enterprise-level data and span other departments within the company.

Legacy or home-grown applications

These systems are company-specific applications catering to express needs. The platforms and the vendors supplying them are many. While each of these individual applications possesses niche capabilities, as a whole they are severely restricted when it comes to “scaling up” to evolving enterprise requirements.

Limitations/gaps in IT context

- ▶ **Multiple functional systems**—Many companies have to deal with multi-CAD, multi-PDM systems within the same organization, where the platforms or formats they support may not be the same.
- ▶ **Lack of open standards and scalability**—Some of these systems don't support existing open standards or scale very well to support enterprise-wide integration, meaning they may not support the future needs of the business.
- ▶ **Information islands**—Here data is isolated and available only in the native product. Data migration and the portability of existing data is thus a huge and expensive task, and companies may not want to port to a new system due to the considerable effort involved in changing the skill set of its workforce.
- ▶ **Lack of federator or process orchestration**—Most PDM and PLM systems have standalone workflows internal to these applications, whereas business processes have workflows that span multiple applications and users.
- ▶ **Lack of collaborative tools**—One of the business drivers is to support collaborative environment in order to facilitate collaboration-intensive tasks. Some companies have invested in tools that enable them to do just that, yet that are not integrated or automated to the tasks they perform. These usually exist within an independent IT infrastructure.
- ▶ **Lack of enterprise-wide security framework**—Most applications have some level of security, usually through a user ID/password. This leads to proliferation of user IDs and passwords for each individual system. Rarely do we find an enterprise-level security model that supports single sign-on, role-based authentication, and access control. There is a strong need for control and consolidation of access rights.

⁶⁷ Product Lifecycle Management

⁶⁸ Enterprise-wide Resource Planning

- ▶ **Redundant data**—Businesses often have identical data managed in overlapping systems (especially legacy systems) that cater to different business processes. Usually these systems are manually updated and hence prone to errors and inaccuracies. These can be automated using process-to-process communication to avoid manual synchronization.
- ▶ **Lack of EAI framework**—Integration efforts if any are isolated and point-to-point integrations. Lack of a standard EAI framework to support present and future needs can lead to poor maintainability.

5.2 The solution

Catering to business needs, the Galaxia solution was formulated using the following methodology:

- ▶ The Galaxia team looked at a generic set of requirements and identified the generic use cases that addressed these.
- ▶ It tried to match these generic use cases to those of a standard pattern (in this case, the Portal composite pattern), hoping that simply customizing this pattern would cover the remaining use cases.
- ▶ It then built a Solution Overview Diagram based on these generic use cases to help identify the business and application integration patterns. This gave a good starting point for identifying those applications that could satisfy the remaining requirements or estimate the development effort required to cover those requirements.
- ▶ This analysis resulted in a “drill down” for identifying the runtime patterns, and finally the product mapping to cover the functional and non-functional requirements.

5.2.1 Requirement summary

Key requirements included:

- ▶ A Web-based solution supporting the extended enterprise and minimizing deployment efforts
- ▶ Harnessing the existing infrastructure and applications
- ▶ Fine-grained access control to enterprise applications
- ▶ Unified security and single sign-on
- ▶ BPM⁶⁹ through a process/work-centric EAI in order to enable integration and process mapping
- ▶ Process visibility and process management tools (monitoring)
- ▶ A unified workplace and personalization

⁶⁹ Business Process Management

- ▶ Web access to engineering data such as part drawings, product structure, product metadata, and other design-related content
- ▶ Sharing design information through workflow
- ▶ Paperless communication
- ▶ Collaborative tools
- ▶ Role-based authorization to view applications and data
- ▶ Support for open standards and industry-proven technology
- ▶ Specified reports and audits
- ▶ An aggregated view of information

The generic requirements spelled out the need for a *Dynamic Workplace* providing the extended engineering community with the ability to work together in a business process by providing a secure and standard framework, and where the solution should scale to support existing/new business processes and multiple business processes.

5.2.2 Generic use cases for the solution

Based on these requirements, the team now had the set of generic use cases listed in Table 5-1 on page 214.⁷⁰

⁷⁰ Generic use case covered by the Portal composite pattern. Use cases covered by standard Portal composite pattern are shown in blue. Those shown in red are the delta use cases.

Table 5-1 Generic use cases

<ol style="list-style-type: none">1. User Registration—The process of registering a user to access the system. Here user could be either internal users or extended enterprise (customers, suppliers, or partners).2. Authentication—Validation of the user accessing the system.3. Authorization—Control of access to applications and contents based on user privileges. The extended enterprise might not have access to all applications or data, with the manager or supervisor level perhaps having special access to certain applications and/or reports, as well as the ability to assign tasks, schedule meetings, etc.4. Personalize Home Page—Personalized access and view of content.5. Access to Enterprise Applications—Aggregated view of enterprise applications such as ERP systems, PDM systems, or legacy/custom-built Web applications. Access to such applications should be role-based, with single sign-on access to enterprise applications.6. View and Send e-Mail—e-Mails to capture initial change request from users and to send notifications about upcoming meetings and new or adjusted schedules.7. Manage Users—Administration of registered user privileges. Add, delete, or modify users.8. Publish and Manage User Content—Personalized-view access to applications and data.9. Synchronous Collaboration—Chatting and whiteboarding to collaborate and share. For example, Engineering Change Requirements may have to be defined through formal interactions between the design lead/manager, the request initiator, etc. Suppliers and partners participate in team and review meetings.
<ol style="list-style-type: none">10. Process/Activity Based Workflow—The change request process involves collaboration and approvals from multiple entities. Information required in the process spans multiple activities and is available within different time lines.<ol style="list-style-type: none">a. Create ECR—Trusted-user requests for a change in a RFC (Request for Change) or ECR (Engineering Change Request) process.b. Approve ECR—Managers or team leaders approve or reject an RFC or ECR.c. Manage and Schedule Work—Managers or team leaders require managing and scheduling capabilities such as transferring, suspending, or terminating an ECR. Suppliers and partners should be able to exchange information with the business in order to realize an ECR.

11. **Workflow Monitoring**—Managers need process monitoring capability to have better process visibility.
12. **Trigger/Control Application Level Workflows**—Business Process Automation through federating existing enterprise applications. For example an approved ECR should automatically trigger a creation of actions in a PDM system. At the end of the design change, the design is versioned, released, and stored in a PDM repository.
13. **Work Notifications and e-Mails**—Collaborate and automate e-mail notifications for change requests or change approval requests.
 - a. Notifications to serve as reminders to do a task
 - b. Suppliers and partners to be notified regarding the changes
 - c. Task delay or base line date expiry notifications
14. **Access to Multiple Business Processes**—Users can belong to multiple business processes, with unified access to all processes to help them manage work efficiently.
15. **Attach Documents to Workflows**—Ability to attach documents to a process workflow to provide additional information during design changes.
16. **Object Representation of Processes and Linking**—Ability to view process data in terms of objects and to see relations to other process objects. For example, one or more ECRs might result in an ECO creation.
17. **Create and Manage Business Object**—Model business entities as objects with dependencies, showing relationships and inheritance for a better view of the business model.
18. **Consolidated View**—Consolidate and view information from various data stores to help users in decision making.
19. **Create and Deploy Business Processes**—Create and deploy process models.
20. **Customizable Reports and Audit Trails**—Reports and audit trails to help monitor current and previous processes. Managers have a global view of every process. The individual user views customized reports based on his or her preferences and settings.
21. **Access to Intranet and Syndicated Content**—Users view/access corporate intranet applications and content.
22. **Access Engineering Data to Build Design Context**—The user reviews part drawings, affected product structure, and change metadata—then builds design context, makes design changes, and notifies others of the change.

5.2.3 Solution Overview Diagram and identification of patterns

The Solution Overview Diagram shown in Figure 5-4 focuses on the ECR process.

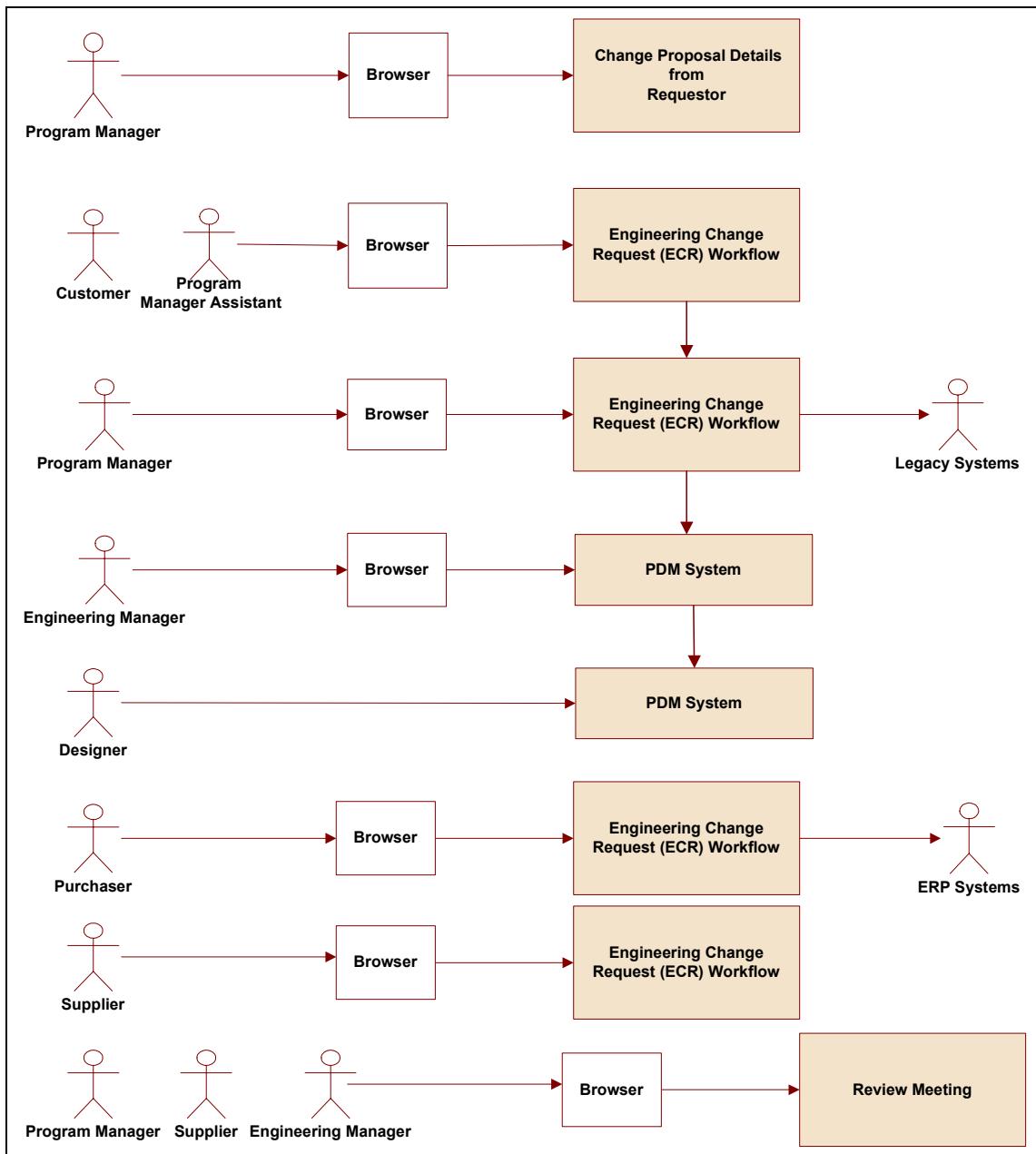


Figure 5-4 Solution Overview Diagram

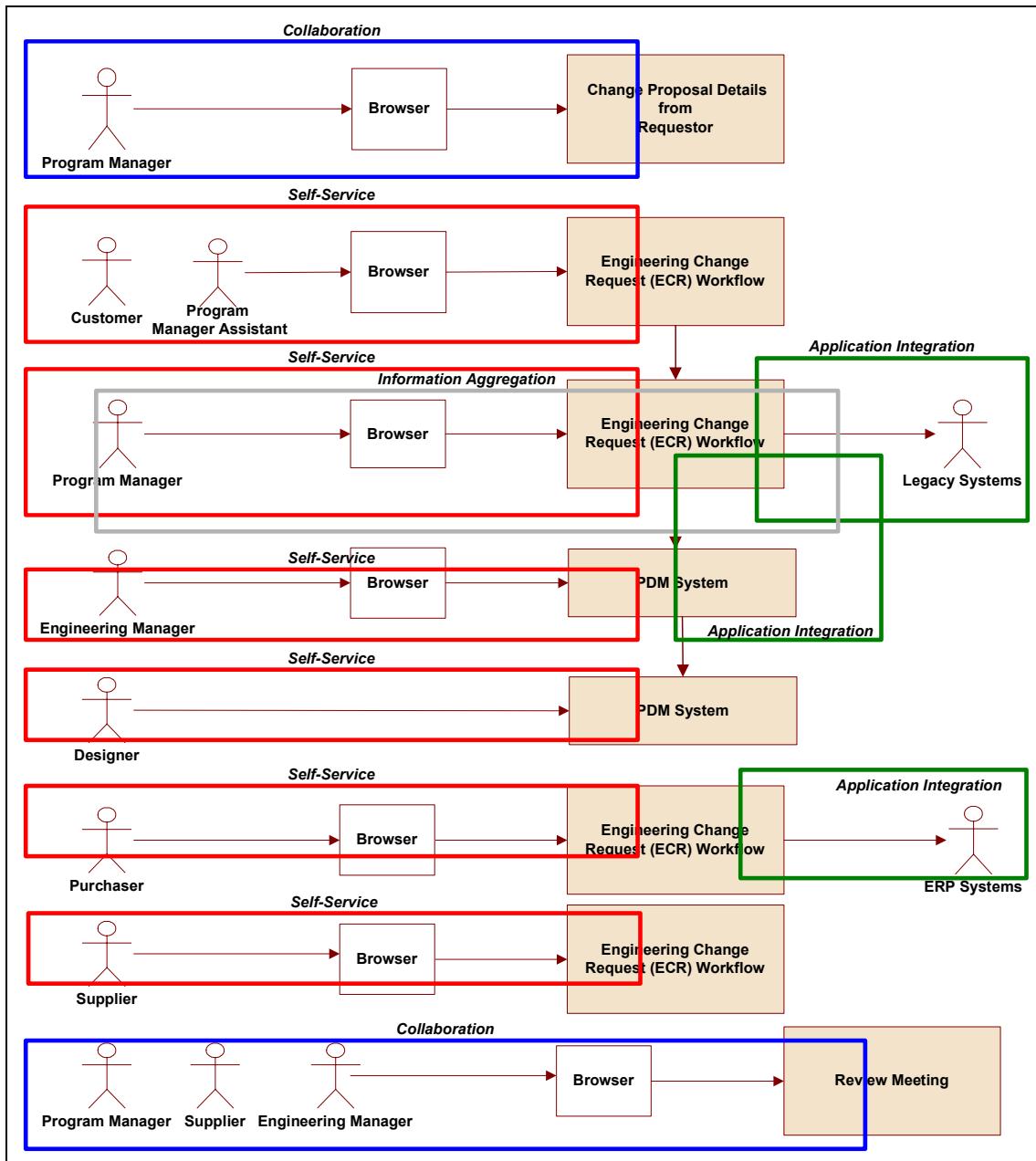


Figure 5-5 Apply business and integration patterns to the Solution Overview Diagram

The Solution Overview Diagram with the functional blocks and patterns shown in Figure 5-5 tells us we need an environment that will enable existing applications

with new functional needs like collaboration, aggregation, and holistic views of information from different applications, direct access to native applications, etc.

The Non-Functional Requirements for this solution include “ilities” like *scalability*, *availability*, *maintainability*, and *manageability*. The other NFRs include security features like *authentication and authorization*, *personalized view of content*, and *managing content*.

Figure 5-5 on page 217 brings clarity in terms of justifying the need for a portal solution

Why a portal solution?

Rationale: When the team made a comparison of the required solution's generic use cases with that of an existing asset like Portal, they saw that some of the Non Functional Requirements like security (authentication and authorization) are covered by the portal. The ease with which users and content can be managed is another positive argument for a Portal-based solution. These features are supported “out of the box” by a portal solution, thereby reducing development effort required to provide these essential functionalities. Many clients have complementary reusable assets (example—collaboration tools) that can easily be integrated with the portal. The Solution Overview Diagram shows the various actors performing assorted business functions using a Web interface. It would be beneficial if these interfaces/applications could be collated/aggregated to provide a toolset for performing the user's tasks.

The portal is standards-based and can scale well to include future requirements.

The Solution Overview Diagram shows that the business patterns do not occur independently. Application integrations are required to satisfy other functional requirements such as triggering workflows in other applications based on business rules/policies, view of data, and use of data from other applications/data sources for effective decision-making.

5.2.4 Identification of Application patterns

The team next tried to identify application patterns which would be a good fit to cover these delta use cases. ePCM uses a process-focused Application Integration application pattern called *Managed Process*. This process-centric Integration pattern is required when Business Process Management is needed to drive an extended business process. This design is used for long-running business processes that often involve complex high-latency applications where the process can extend anywhere from minutes to days in duration.

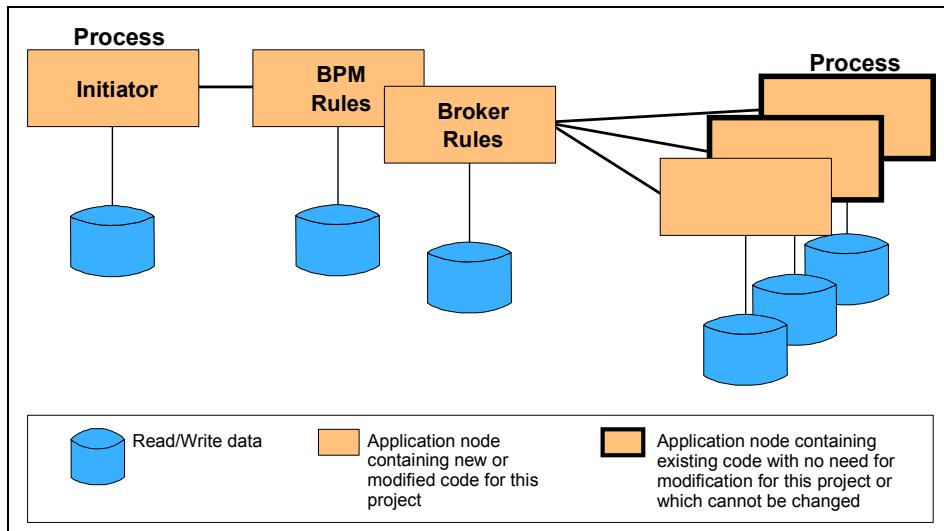


Figure 5-6 Managed Process pattern

Business Process Management extends the brokered approach to application integration with extended workflow- and process-state persistence services. Sophisticated recovery and exception handling can be implemented in the BPM rules system using process modeling tools. BPM provides an easily defined layer on top of existing processes to coordinate the execution of each individual operation. BPM coupled with the integration broker enables the long-running process to initiate each step through the integration broker to take advantage of its routing and transformation capabilities.

The integration broker along with the BPM rules (process templates) federate other enterprise applications like ERP systems, PDM systems, or legacy applications. This pattern allows triggering of existing application workflows, with end-to-end process integration.

- ▶ The PCM⁷¹ process requires users to access multiple applications and information sources to finish a unit of work.
- ▶ Supplier and partner applications should be integrated in the change management process to synchronize design/production schedules and shorten product development cycle time. Here the extent of collaboration between business processes depends on the relationship between the businesses.
- ▶ Individual applications have workflows that must be orchestrated rather than data consolidation.

⁷¹ Product Change Management

- ▶ PCM process latency is quite high and can last anywhere from minutes to days.
- ▶ The application portfolio involved in the PCM process is diverse. It includes a mixture of pre-packaged software and homegrown custom applications.

Hence the Managed Process application pattern is a good fit to integrate complex business processes and applications.

Key benefits

- ▶ Re-use of investment in existing enterprise applications/assets
- ▶ End-to-end process visibility
- ▶ Use of ePCM functionality, leading to better control over business processes
- ▶ Quick implementation and faster ROI

Customized presentation to host⁷²

Here the pattern discussion becomes specific to the application in terms of visualizing and accessing data normally reserved to a specific set of users. Engineering data and metadata about drawings, etc., are stored in PDM systems. This data must be shared and accessed by both engineering and non-engineering users to enable better decision-making in the business process flow. The key driver here is to expose an existing application to a wider audience. Products such as ENOVIA 3d com Navigator can be used to enable Web access for such data, thereby making use of existing assets. ENOVIA 3d com Navigator provides Web access to ENOVIA VPM (see “Sidebar: ENOVIA Products” on page 221) data that includes product structure, configuration, associated part drawings, and engineering metadata. This data can be both viewed and modified, provided the user has the required access rights.

For example, an ECR request correlates to the engineering data in a PDM/PLM system. This data can be the product structure, associated assembly/part drawings, documents, reports, etc. The drawings and documents related to an ECR can be viewed by managers and team leaders to review and help them make decisions, thereby reducing the latency of business events.

⁷² Also known as U2B Topology 4



Sidebar: ENOVIA Products ^a

ENOVIA's e-business solutions for collaborative product data management and decision support are marketed, sold and supported by IBM and its business partners, and are a key component of the overall PLM set of solutions offered by Dassault Systèmes, a world leader in PLM with more than 65,000 customers in 80 countries.

ENOVIA 3d com Navigator

ENOVIA 3d com Navigator provides a single front-end to multiple information sources and applications for improving collaboration and decision-making. ENOVIA 3d com Navigator (N3G) accesses, visualizes, and publishes data stored in multiple ENOVIA PDM systems. ENOVIA 3d com Navigator (N3G) can be implemented as a set of modular components (portlets) that can be implemented within a WebSphere Portal environment, enabling organizations to build their own collaborative E2E or B2B business portals.

Because they leverage both Web and 3D technologies, ENOVIA 3d com Navigator products are the ideal solution for any company needing a single point of access uniting all extended enterprise information. This single point of access facilitates decision making, collaboration, and access to the product lifecycle pipeline.

V5 Virtual Product Modeling (VPM)

ENOVIA V5 VPM provides manufacturing companies with a state-of-the-art virtual product management solution. This allows them to deliver innovative products that can be easily modeled in many variants to meet market requirements without increasing manufacturing complexity in a shorter time to market.

Going beyond current digital mock-up technology that offers only 3D product visualization and navigation, ENOVIA V5 VPM captures and manages the engineering design intent by exposing specifications, engineering rules, operational parameters, simulation results, manufacturing process, and change context, which is key to understanding and evolving the product definition. This rich network of knowledge provides complete understanding of the relationship between a product's components, manufacturing, and maintenance technology. This allows engineers to perform necessary changes to the product design until its performance and cost are optimized. With relational design, change impacts can be easily analyzed and corrected, with change notifications automatically sent to impacted organizations.

a. <http://plm.3ds.com/en/enovia/products/overview.asp>

Both ePCM and ENOVIA 3d com Navigator provide "out-of-the-box" functionality described by delta use cases. In order to expose these applications and capture the U2B⁷³ requirements, Galaxia used the *Customized Presentation to Host* self-service application pattern. Here applications are exposed as portlets to enable users to exploit their features and functionality through a portal. Providing

⁷³ User to Business

access through a portal helps Galaxia take advantage of the portal's security features, single workplace for accessing information/content, and collaborative toolsets.

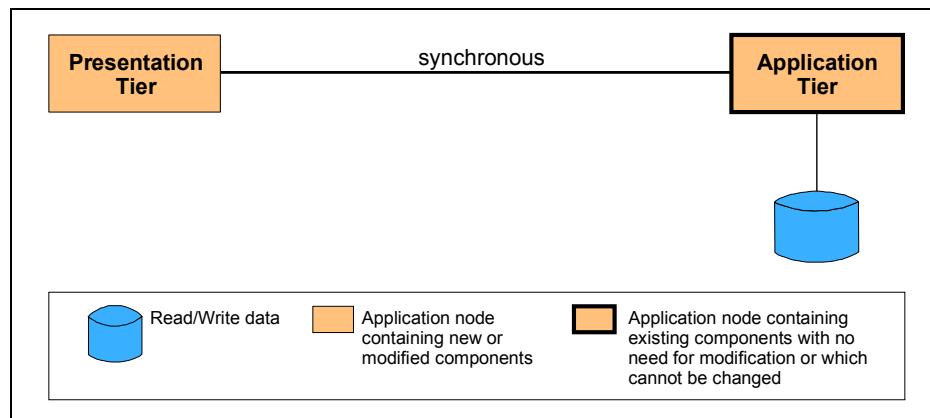


Figure 5-7 Customized Presentation to Host (U2B Topology 4)

For more details on this pattern, refer to *IBM Patterns for e-business*.⁷⁴

Collaboration through Lotus Instant Messaging and Domino®

Lotus Domino and Instant Messaging⁷⁵ portlets can be used to resolve such needs as synchronous and asynchronous collaboration.

Especially in the Automotive and Aerospace industries, the change management process is a highly collaborative process involving different departments within the organization and its extended enterprise (e.g., suppliers and partners), where the organization and its suppliers commonly reside in different geographical locations.

Examples of “user-to-user” interaction include:

- ▶ Engineering Change Requirements having to be defined through formal interactions between the design lead/manager, the initiator of the request, etc.
- ▶ Suppliers and partners having to be notified that they must participate in team and review meetings regarding changes
- ▶ Users needing access to multiple applications and information sources without every application requiring its own sign-on⁷⁶
- ▶ Customized access to contents required for various partners, suppliers, and internal users⁷⁷

⁷⁴ <http://www.ibm.com/developerworks/patterns/>

⁷⁵ Formerly Sametime®

⁷⁶ Implementing single sign-on for applications running on different application servers requires complex modifications and migrations.

Galaxia needed to customize the Portal composite pattern to allow it to have this additional functionality. Hence the Portal composite pattern was customized to add delta application patterns.

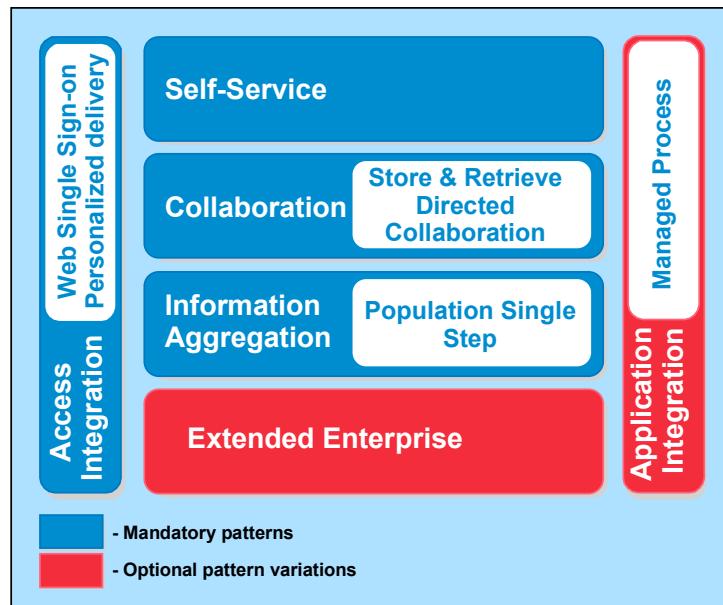


Figure 5-8 Dynamic Workplace—Portal composite pattern variant

5.2.5 Applying the delta Runtime patterns to the Portal composite runtime pattern

The Dynamic Workplace composite pattern is a custom Portal composite pattern that includes the mandatory and optional patterns. The delta Runtime patterns were added to the Portal composite pattern Runtime pattern to get the Dynamic Workplace composite pattern.

⁷⁷ The ability to control access to certain applications or information can be addressed by this pattern.

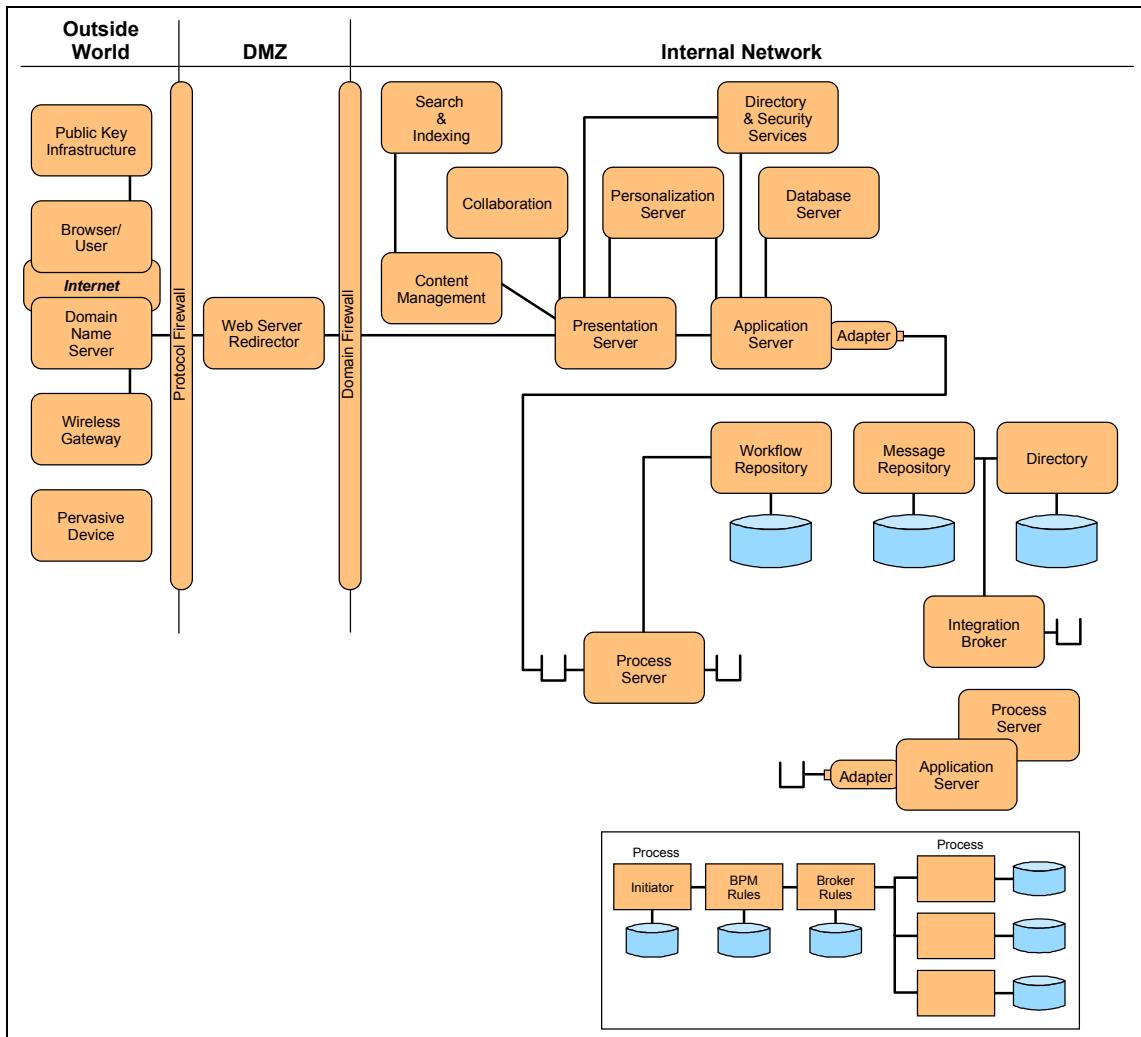


Figure 5-9 Galaxia solution—Runtime pattern

Business Process Management (BPM) can be achieved through process- or work-centric EAI.⁷⁸ Business processes can be visualized as activities that cut across many existing applications and tasks. These individual applications have pre-existing workflows that must be integrated in the overall business process. Many pre-packaged applications contain a workflow solution as part of their functionality.⁷⁹

Therefore, an extended business process like ECR needs to link with the existing workflows as part of its processing. The Runtime pattern depicts the overall architecture based on the Application patterns.

The Runtime pattern is a variation of the standard Portal composite Runtime pattern that includes application integration requirements. It takes advantage of a portal implementation to leverage the concept of personalization, aggregation of disparate applications through a common window, providing access to content and applications based on roles and preference, and a collaborative framework that can be customized to suit business needs. The Runtime pattern includes an application integration framework that caters to such needs as federating workflows in individual applications and global process visibility through better business process management.

The Runtime pattern identifies the functional areas that will likely need to be addressed when considering this type of implementation. This pattern helped Galaxia identify products that could be used as well as how products in the target environment could be used to satisfy business needs.

⁷⁸ Enterprise Application Integration

⁷⁹ For example, SAP R/3, PDM systems like ENOVIA VPM, or Lotus Notes®

5.2.6 Product mapping

To have a quick implementation of the suggested solution in the Runtime pattern, Galaxia needed to look at existing products containing the necessary features. Additional customization of such products and codification might be necessary at times to meet business requirements. Selection criteria for the best-fit product for each function can be based on:

- ▶ Existing systems and platform investments
- ▶ Available customer and developer skills
- ▶ Customer choice
- ▶ Future functional enhancement direction
- ▶ Standards compliance

The products and technologies chosen should fit into the target environment and ensure quality of service (such as scalability and reliability) so that the solution can grow along with the e-business. The products identified for the Portal composite Runtime pattern variant are shown in Figure 5-10 on page 227.

Note: CrossWorlds® is now known as WebSphere Business Integration.

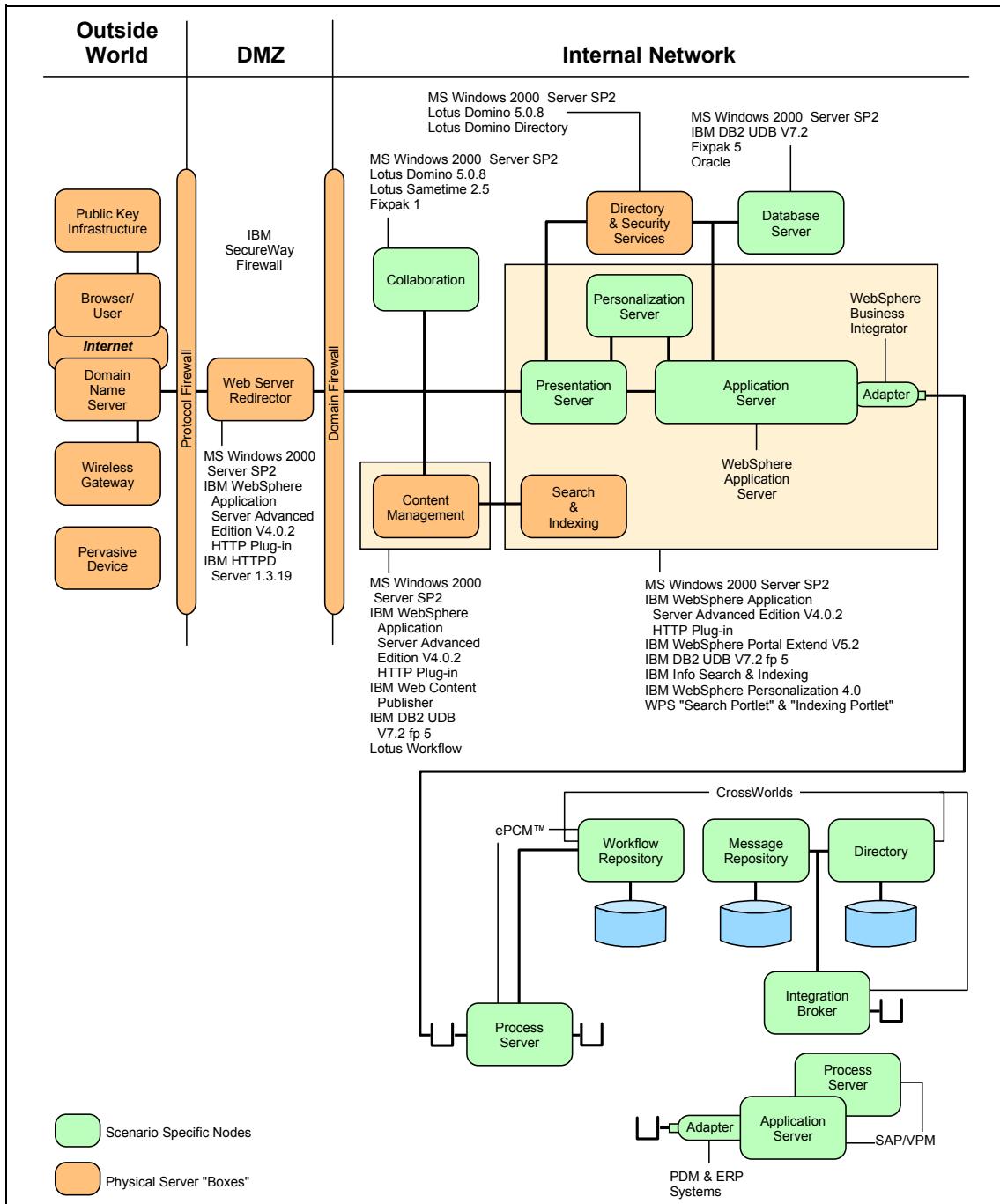


Figure 5-10 Galaxia product map

After drilling down to the Runtime pattern we see the product map, which takes into account the existing applications and platform investments made by the industry.

Businesses have invested in PDM systems to create and store product information and in ERP systems to collect and store enterprise data. Some have made investments in stand-alone content management systems and collaborative tools such as Lotus products, which can be easily integrated and put to use effectively.

- ▶ **WebSphere Portal Server** is an ideal candidate that fits the standard Portal composite pattern and adds “out-of-the-box” functionality, including:

- User registration
- User management
- Authorization
- Authentication
- Single sign-on and security features, enabling standard access to enterprise applications and content
- Personalization
- Publishing and managing content

- ▶ **ePCM** is standards-based and built on IBM's WebSphere Business Integration framework. The underlying products used are WebSphere MQ, WebSphere MQ Workflow, and DB2. It also takes advantage of its synergy with WebSphere Interchange Server EAI platform. ePCM acts as a workflow federator to manage various enterprise-wide processes, integrating enterprise-wide applications like ERP and engineering systems like PDM. Some of the key functionality supported by ePCM includes:

- Enterprise workflow and process monitoring
- Process orchestration and federation, which includes triggering application workflows and updating application data
- Consolidated view of application data to enable better decision making
- Object representation of process data and linking
- Reports and auditing of processes
- Document attachments and workflow
- Workflow and e-mail notifications

- ▶ **Lotus Instant Messaging, Lotus Web Conferencing, and Lotus Domino** are used to enable synchronous and asynchronous collaboration. These products add functionality such as e-meeting and e-mail integration.

Users can share application content using whiteboarding capability. The Instant Messaging chat feature enables synchronous collaboration. Review and team meetings can be conducted using this facility.

- ▶ **ENOVIA 3d com** is a Web-based application used to access engineering data.⁸⁰ Some of the features in this application include:
 - Accessing engineering data such as parts and product structure stored in ENOVIA VPM
 - Building design context, viewing affected parts, and making design changes
 - Defining action flows and consolidating design changes
 - Visualizing DMU

Note: Galaxia used ENOVIA 3d com Navigator to access engineering data. This application is specific to the ENOVIA VPM PDM system. You may have to use other Web-based applications to extract data from other PDM Systems.

5.2.7 On Demand Workplace

This product mapping leads to a solution that aims at being an *On Demand Workplace*.

The advantages of the Dynamic Workplace are:

- ▶ Better-managed processes and efficient work management
- ▶ A single place to do work, thereby improving the productivity of the workforce
- ▶ Reduces process cycle time and latency of business events
- ▶ Personalized content and inclusion of the extended enterprise
- ▶ Paperless and effective communication
- ▶ A decision support system with reports and audit trails
- ▶ User- and role-based access control and authorization, reducing application development time and enabling content management
- ▶ Scalable to include future business integration needs
- ▶ Leverages existing skills
- ▶ Standard “out-of-the-box” features, a portal infrastructure, and ePCM templates leading to quick deployment and faster ROI
- ▶ A strong security framework for applications and data plus a library of available portlets, reducing application development cost
- ▶ WebSphere Portal integrates well with Lotus applications and hence can provide collaborative capabilities with minimum customization effort
- ▶ Quick ROI, aggregating access to existing applications and thereby enabling reuse of current investments
- ▶ Ease of use and administration, leading to greater productivity
- ▶ Lower TCO⁸¹ (encompassing license, infrastructure, implementation, and maintenance costs)

⁸⁰ Refer to “Sidebar: ENOVIA Products” on page 221

The benefits highlighted for the Galaxia solution cover most of the reported IT priorities (see Figure 2-7 on page 51). This solution is a good framework to satisfy the requirements of an “on demand” workplace (see Figure 2-8 on page 60). It can be easily adapted to integrate new suppliers, customers, and IT infrastructure (e.g., PDM and ERP systems).

5.3 The Galaxia demo scenario

Note: In the scenario discussed in this section, the PDM system is ENOVIA VPM, the ERP system is SAP R/3, and the engineering release repository is Teamcenter Enterprise. Other equivalent products or solutions could very well have replaced these systems. SAP R/3 is a product of SAP AG. Teamcenter Enterprise, previously known as Metaphase, is a product of UGS PLM Solutions, an EDS company.

The demo showcases the ECR⁸² process, which manages design-related engineering issues. This process involves many users and sharing of information that exists in multiple systems.

5.3.1 Steps in a typical ECR process

1. A typical ECR can be initiated by a request from the customer, supplier, partner, or just about anyone else involved with the product.
2. The project manager validates the ECR and approves the design change request.
3. The ECR is then sent to the design department to undertake the necessary changes. The engineering manager builds the context for analysis using the given details and approves an action using a PDM system. The action is then transferred to the designer to make the necessary design change.
4. The designer makes the proposed design and then sends the design details to the engineering manager for approval.
5. The engineering manager approves the design and sends it back to the project manager.
6. The project manager then reviews the design change along with his team. He may require additional information from other enterprise systems to make his decision.
7. If the design is approved, it results in an ECO⁸³ process.

⁸¹ Total Cost of Ownership

⁸² Engineering Change Request

This illustrates the general outline of an ECR process. Many businesses have some sort of custom-built application to manage and keep track of their ECRs, while others have a paper-based workflow to manage them. The potential pitfalls in such systems are:

- ▶ Long process cycle times
- ▶ Lots of paper and loss of valuable knowledge
- ▶ Lack of collaboration
- ▶ Missing information or lack of the data visibility required to make early decisions
- ▶ Information required to carry out the design change resides in multiple systems
- ▶ Collaborating and involving the partners and suppliers in the ECR may not be seamless

The demo shows the benefits of an On Demand Workplace using a portal as a single point of personalized interaction with the applications, content, processes, and people.

Table 5-2 Applications and users involved in the scenario

Role	Designer	PM Assistant	PM Manager	Engineering Manager	Purchaser	Supplier	Client
Lotus Notes	X	X	X	X	X	X	X
Lotus Instant Messaging	X	X	X	X			
WebSphere Portal	X	X	X	X	X	X	X
ePCM	X	X	X	X	X	X	
ENOVA VPM	X	X	X	X	X		
ENOVA 3d com Navigator	X	X	X	X	X		
Teamcenter Enterprise	X	X	X	X			
SAP/R3	X	X	X	X			

Table 5-2 shows the users and the applications they have access to.

⁸³ Engineering Change Order

5.3.2 Product mapping for the scenario

Product mapping shows the applications and the part they play in the scenario.

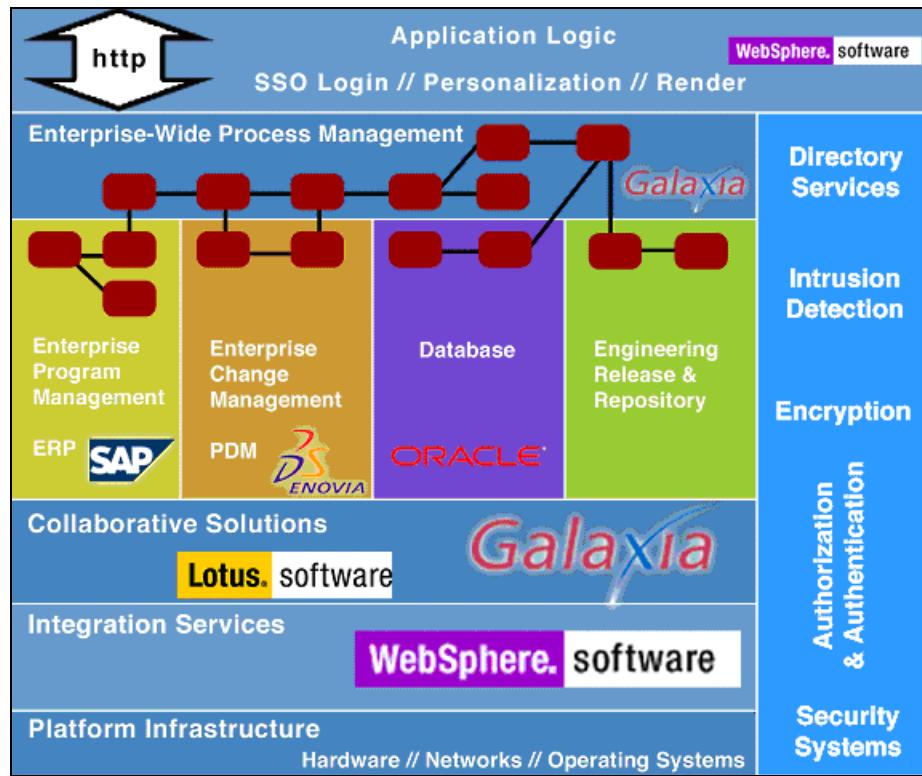


Figure 5-11 Scenario product mapping

The applications involved in the scenario are:

- ▶ **ePCM**—Business Process Manager and federator of enterprise applications
- ▶ **ENOVIA 3d com Navigator//ENOVIA VPM**—A VPDM system that manages the design contents and product definitions
- ▶ **SAP R/3**—Enterprise Resource Planning system
- ▶ **WebSphere Portal**—Unified and personalized workplace
- ▶ **Lotus Instant Messaging and Web Conferencing**—Collaboration through e-meetings and instant messaging
- ▶ **Lotus Notes**—Collaboration and e-mail
- ▶ **Teamcenter Enterprise**—PDM system used as an Engineering Release Repository

Note: Figure 5-12 was adopted from Figure 5-13 on page 234.

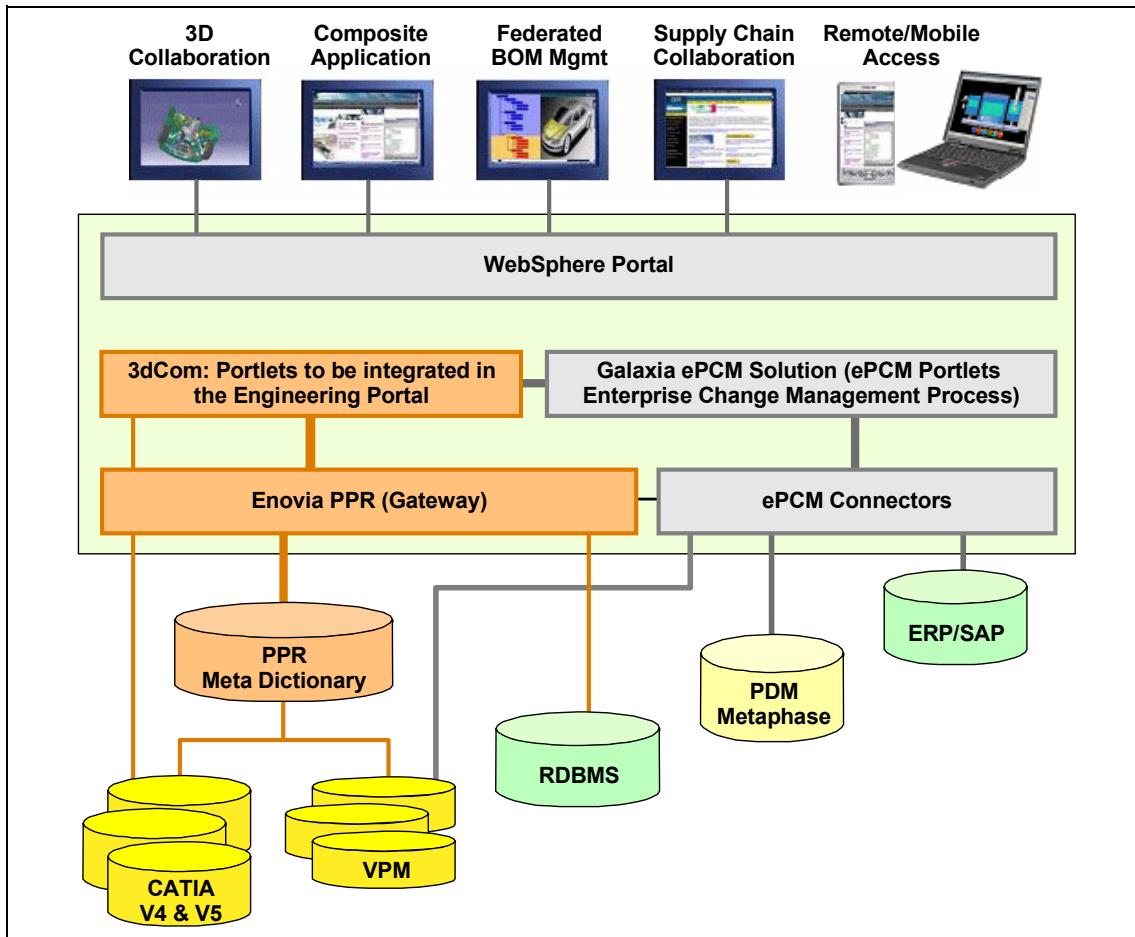


Figure 5-12 Functional architecture

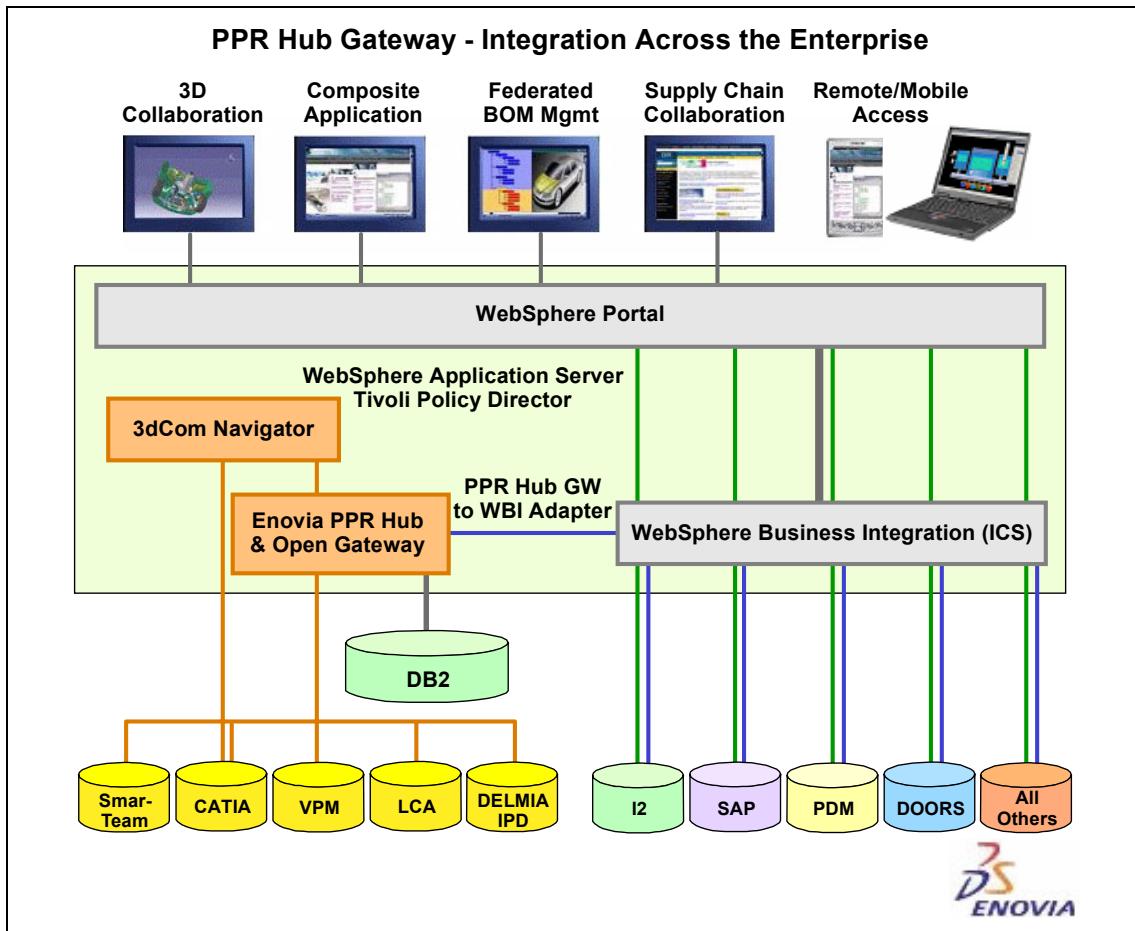


Figure 5-13 Dassault Systemes/IBM PLM architecture chart



Sidebar: CATIA ^a

CATIA is the product design solution in the PLM offering from Dassault Systèmes. It allows manufacturers to simulate all of the industrial design processes—from the pre-project phase through detailed design, analysis, simulation, assembly, and maintenance.

CATIA Version 5 is an integrated suite of Computer Aided Design (CAD), Computer Aided Engineering (CAE), and Computer Aided Manufacturing (CAM) applications for digital product definition and simulation. Modular in design, the CATIA product line adapts to all customer businesses—including style and form design, mechanical design, systems and equipment engineering, managing digital mock-ups, numerical control, simulation, and analysis—using an open and scalable V5 architecture.

a. http://www.3ds.com/en/brands/catia_ipf.asp

5.3.3 Interactions between various applications in the scenario

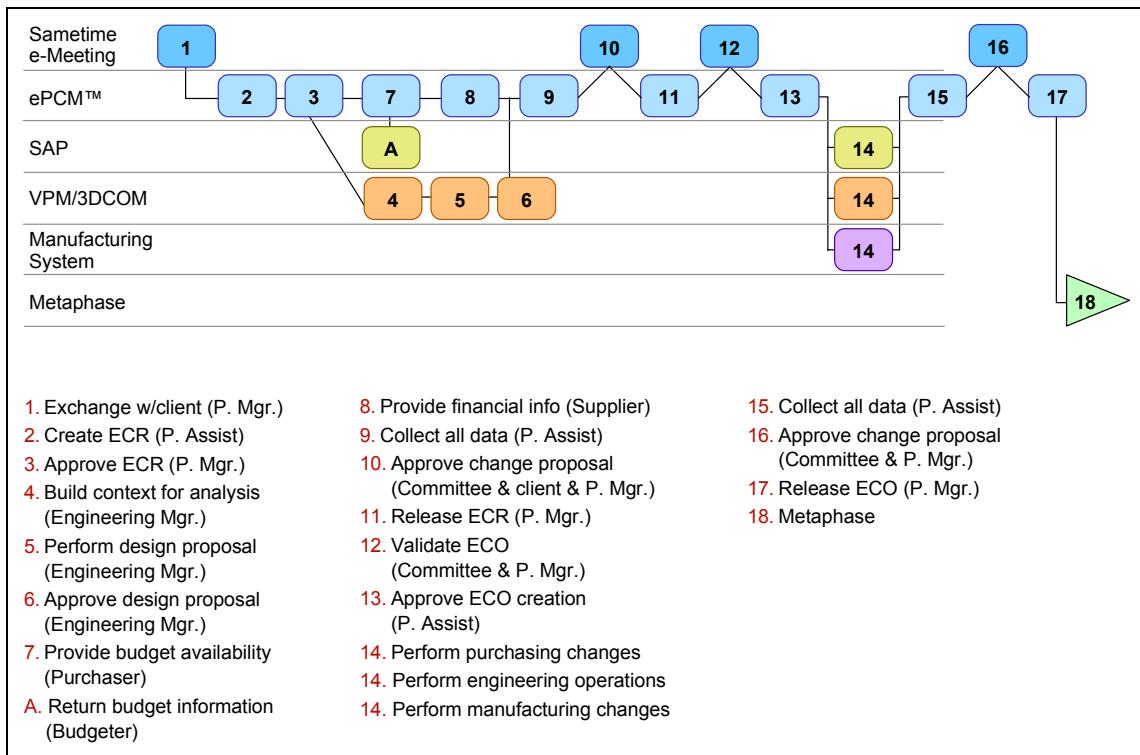


Figure 5-14 Scenario overview⁸⁴

⁸⁴ Metaphase is now known as Teamcenter Enterprise

5.3.4 ECR process scenario

The ECR process scenario has four phases, as shown in Figure 5-15 and described beginning with Table 5-3 on page 236.

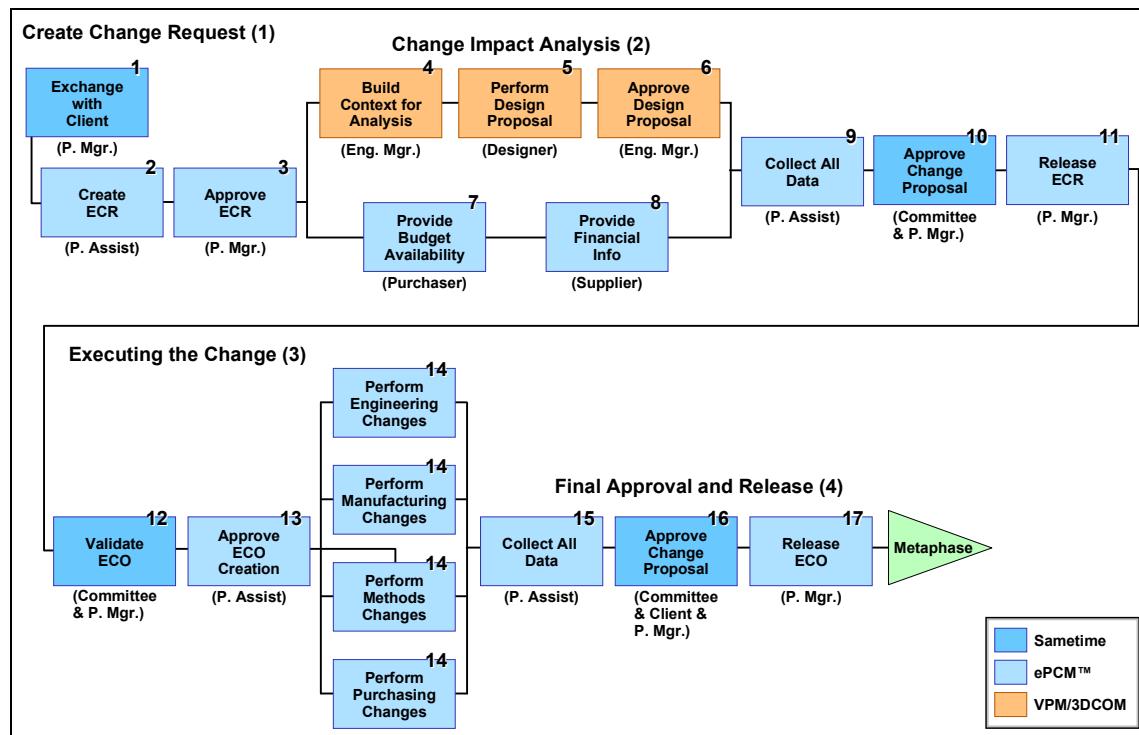


Figure 5-15 Scenario phases

Table 5-3 ECR process phase 1

Phase 1: Identifying the change proposal

1. **Exchange with client**—The Program Manager receives the client's request for change by e-mail or through an e-meeting. The Program Manager gathers the change proposal details to create an accurate ECR. The details collected may include a general description of the change request, the change priority, the change schedule, the change category, parts affected, and reference drawings.

The Program Manager uses Lotus Instant Messaging Portlet to schedule e-meetings and Lotus Notes Mail Portlet to access e-mails.

2. **Create an ECR**—A Program Manager Assistant creates an ECR in ePCM by filling in the ECR details collected in the previous step. The Program Manager Assistant belongs to a role called *Requestor*. Any person belonging to the Requestor role can create an ECR. After creating the ECR, the Program Manager is notified via e-mail that a new ECR was filed and awaits his approval.

Here access integration to multiple applications through the use of portlets is demonstrated.

3. **Approve ECR**—The Program Manager uses the Portal e-mail portlet to check his mail. The Program Manager verifies and validates the ECR filed in the previous step by browsing his worklist in the ePCM portlet. The Program Manager bases his decision to accept or reject the ECR based on information filed in the previous activity and coming from legacy applications. On approval, the ePCM ECR workflow triggers an equivalent workflow in ENOVIA VPM.

This step demonstrates application integration using ePCM and WebSphere Business Integration framework.



Figure 5-16 Create an ECR

Table 5-4 ECR Process Phase 2—Change Impact Analysis—Engineering

Phase 2: Change Impact Analysis—Engineering

4. **Engineering Manager**—The ePCM ECR initiates an action flow in ENOVIA VPM. The Engineering Manager uses ENOVIA 3d com Navigator to access ENOVIA VPM and build a context for analysis. He attaches the relevant part, product structure, and drawings and approves the action. ENOVIA 3d com Navigator provides Web access of ENOVIA VPM data so that the manager can visualize affected part drawings and product structure.

Action is automatically transferred to the Designer in ENOVIA VPM.

This step demonstrates Web-based access to engineering data through an ENOVIA 3d com Navigator portlet.

5. **Designer**—Creates design proposal using native CAD applications and approves action.

Action is automatically transferred to the Engineering Manager.

6. **Engineering Manager**—The Manager views the design changes via 3Dcom and approves the design proposal by completing the action.

A trigger is sent to e-PCM to move on with the workflow in ePCM.



Figure 5-17 Change impact analysis—Engineering

Table 5-5 ECR Process Phase 2—Change Impact Analysis—Engineering

Phase 2: Change Impact Analysis—Finance

7. **Purchaser**—Following ECR approval, ePCM sends ECR details to the purchaser for financial approval. The purchaser receives budget information for the change from SAP and sends a request to the supplier to get time and cost information.

The ECR is automatically pushed to the supplier's worklist.

This step demonstrates application integration capability with an ERP system.

8. **Supplier**—The change may be related to a part provided by a supplier. The supplier is notified of the change request. The supplier provides its information and approves the ECR.

The ECR is automatically pushed to the Program Manager Assistant's worklist.

This step demonstrates the inclusion of the extended enterprise in the ECR process.

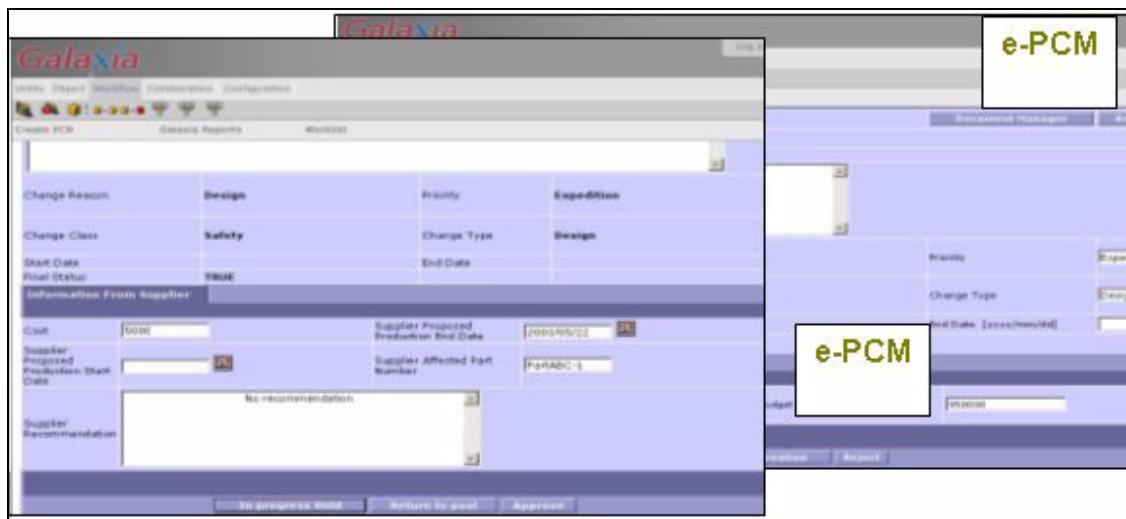


Figure 5-18 Change impact analysis—Finance

Table 5-6 ECR Process Phase 2—Change Impact Analysis—Approval

Phase 2: Change Impact Analysis—Approval

9. **Project Manager Assistant**—The change may be related to a part supplied by a supplier. The supplier is notified about the change request. The supplier provides its information and approves the ECR.

[The ECR is automatically pushed to the Program Manager Assistant's worklist.](#)

This step demonstrates the inclusion of the extended enterprise in the ECR process.

10. **Project Manager**—The Project Manager has a meeting with the committee to review the change (designer + supplier). The Program Manager uses the Lotus Instant Messaging portlet to conduct the review meeting.

11. **Project manager**—Based on satisfactory results of the review meeting, the Project Manager gives final approval for the proposed change and releases the ECR. The ECR process automatically triggers the ECO process in ePCM.

[An ECO is automatically created when the ECR is released.](#)

This step demonstrates inter-process communication using ePCM.

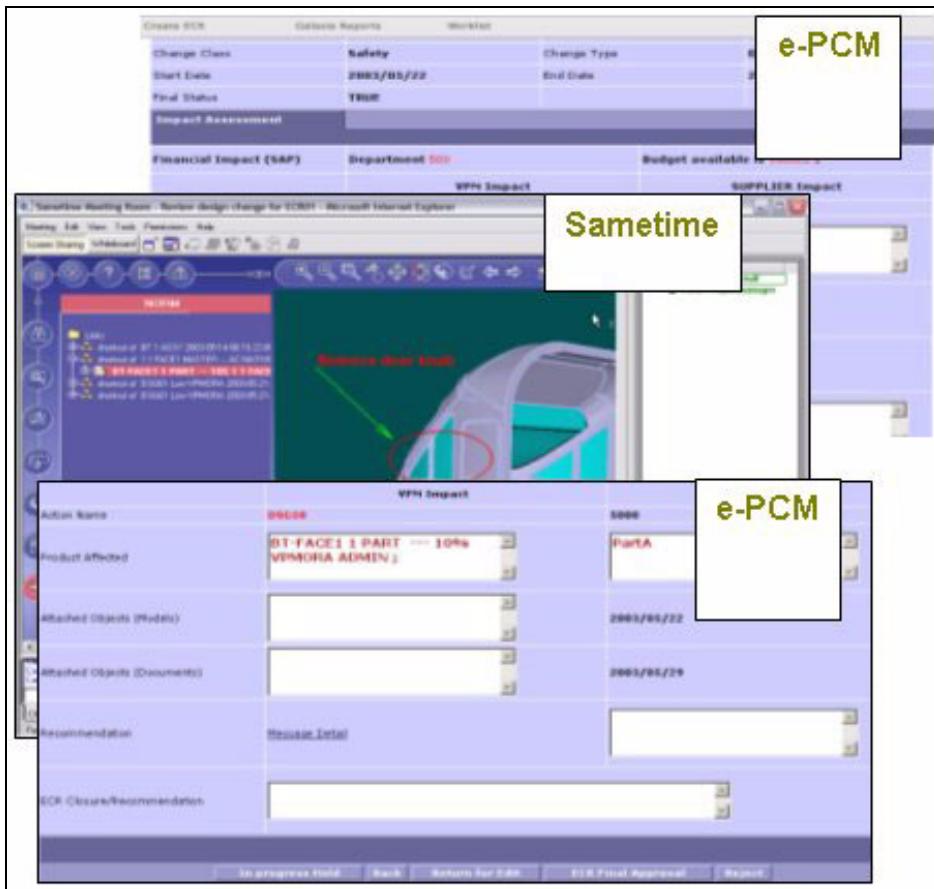


Figure 5-19 Change impact analysis—Approval

Table 5-7 ECR Process Phase 3

Phase 3: Executing the change

12. **Committee**—Using Lotus Instant Messaging, the Committee discusses the plan for the ECO
13. **Project Manager**—Approves the ECO creation
14. **Engineering, Manufacturing, Purchasing, Methods**—Receive notification to perform the change, perform appropriate changes, and approve the change request.
15. **Project Manager Assistant**—Receives data and approvals from different groups

Table 5-8 ECR Process Phase 4

Phase 4: Final review and release

16. Committee and Client—Review the change in DMU

17. Project Manager—Approves ECO creation and forwards ECO details to Teamcenter Enterprise.

Teamcenter Enterprise integration is not demonstrated in this scenario.

5.4 Galaxia ePCM

In the engineering domain, ePCM allows those who participate in the manufacturer's product change management process to work in concert to develop, deliver, and support best-in-class products. ePCM supports everything from simple document-based workflows to workflows with application-level integrations.

Galaxia provide solutions for

- ▶ Engineering
- ▶ Project and program management
- ▶ Procurement and materials
- ▶ Quality management processes
- ▶ IT processes

ePCM supports everything from simple document-based workflows to workflows with application-level integrations.

Minimal training will enable IS/IT people to use the ePCM toolset to model new processes and manage them independently. In partnership with IBM, Galaxia also offers hosting solutions to augment existing IT infrastructures. In the engineering domain, ePCM allows those who participate in the manufacturer's product change management process to work in concert to develop, deliver, and support best-in-class products.

5.4.1 ePCM product overview

Business Process Management involves understanding the process, creating a process model, and integrating people and applications to provide a platform for the customer to optimize and improve its business processes.

ePCM is a process-centric EAI⁸⁵ framework that enables businesses to manage their existing and future business processes. ePCM integrates people and

applications through a user-oriented workflow and application integration framework.

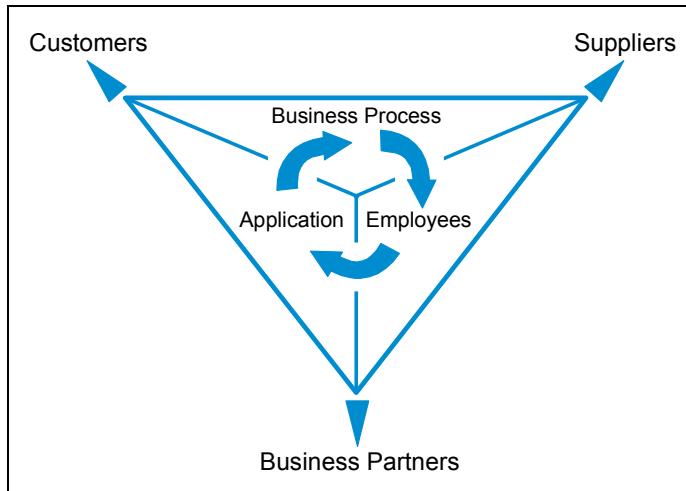


Figure 5-20 EAI framework

It spurs application integration using IBM's WBI⁸⁶ standards and products, and can enhance collaboration through workflows and collaborative tools like Lotus Instant Messaging and Team Workplace.⁸⁷ When integrated with WebSphere Portal, ePCM provides an on demand dynamic workplace for people to collaborate and participate in enterprise-wide business processes.

ePCM architecture follows the standard MVC⁸⁸ pattern and three-tier architecture, that separates presentation, business, and data access logic. The business logic is basically encapsulated in the workflow-based process model. Application integration is based on messaging and uses WebSphere MQ⁸⁹as its backbone.

⁸⁵ Enterprise Application Integration

⁸⁶ WebSphere Business Integration

⁸⁷ Formerly Lotus QuickPlace

⁸⁸ Model-View-Controller

⁸⁹ Formerly MQSeries

Some of the key features of ePCM include:

- ▶ **Process modeling**—The typical process model has many activities and associated users. Access is controlled by role-based authorizations. A process can be modeled to spawn a sub process or trigger an external application. The activities modeled in the process can be manual or automatic, synchronous or asynchronous, and hence can trigger external applications. Users will be notified about work through email via a work item in their worklists. Processes can have any number of conditions or business rules attached, with any number of users and roles, parallel paths, loops, and nested loops.
- ▶ **Process and form templates**—Processes are modeled as standard templates to reduce the build/deployment time of a process model. Use of templates to model and standard forms to view the application data helps business analysts and the IS/IT staff to easily configure a new process-centric application. With minimal training, users can deploy new business processes as well as create and set parameters for the forms required to view the application.

Standard templates modeled in ePCM include:

- Engineering Change Management processes such as RFCs,⁹⁰ ECRs,⁹¹ and ECOs⁹²
- Work Management and Task Allocation processes
- Quality Management processes
- Procurement and Materials
- IT processes

Form templates support automatic ID creation, links to utilities such as Document Manager, support for features like WIP⁹³ notification, and reject loop support.

Worklist templates give detailed information regarding each work item to help users quickly identify the state of a work item.

Drill-down capabilities exist for detailed view or tracking of work item details.

⁹⁰ Requests for Change

⁹¹ Engineering Change Requests

⁹² Engineering Change Orders

⁹³ Work in Progress

5.4.2 Templates for engineering processes

Standard templates for Change Management processes like RFCs, ECRs, and ECOs enable manufacturers to have better control over their product design and to manage design changes.

These templates provide a fast track for implementing a customer-specific solution, leading to tighter cost controls and quality products.

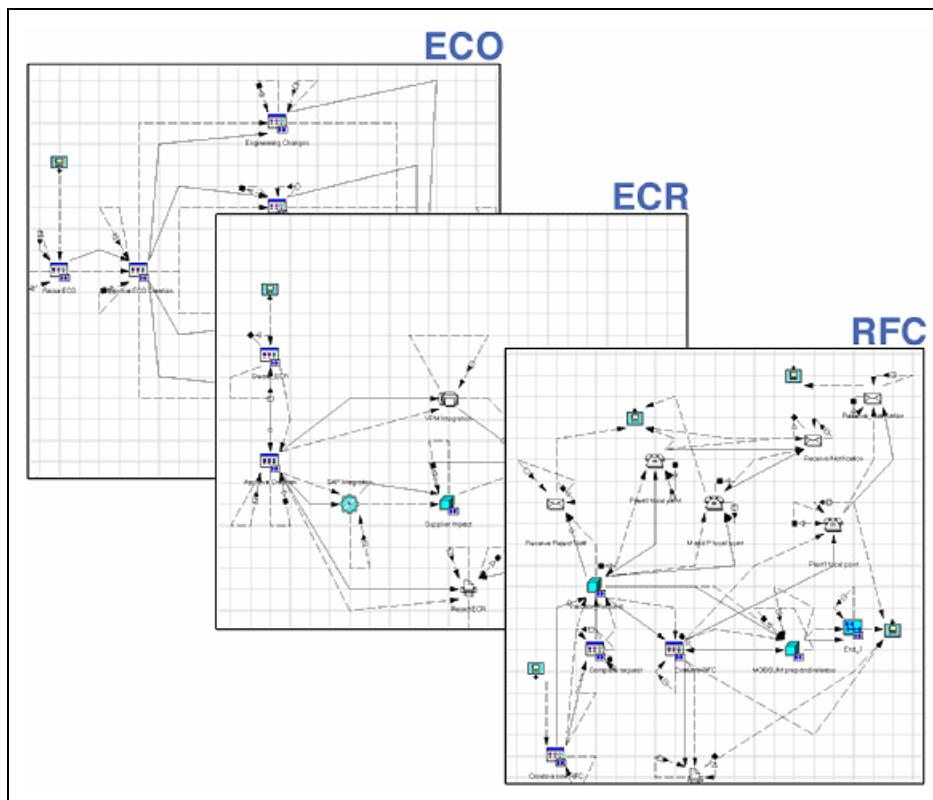


Figure 5-21 Process templates for engineering

5.4.3 Templates for the Work Management and Task Allocation process

The Work Management and Task Allocation template helps managers efficiently track programs, projects, and tasks. This could require multiple levels of management approvals. Risk and Project Management from initialization to completion can be realized using this template. IT/IS managers can use it to manage their IT resources (support staff, project resources, etc.).

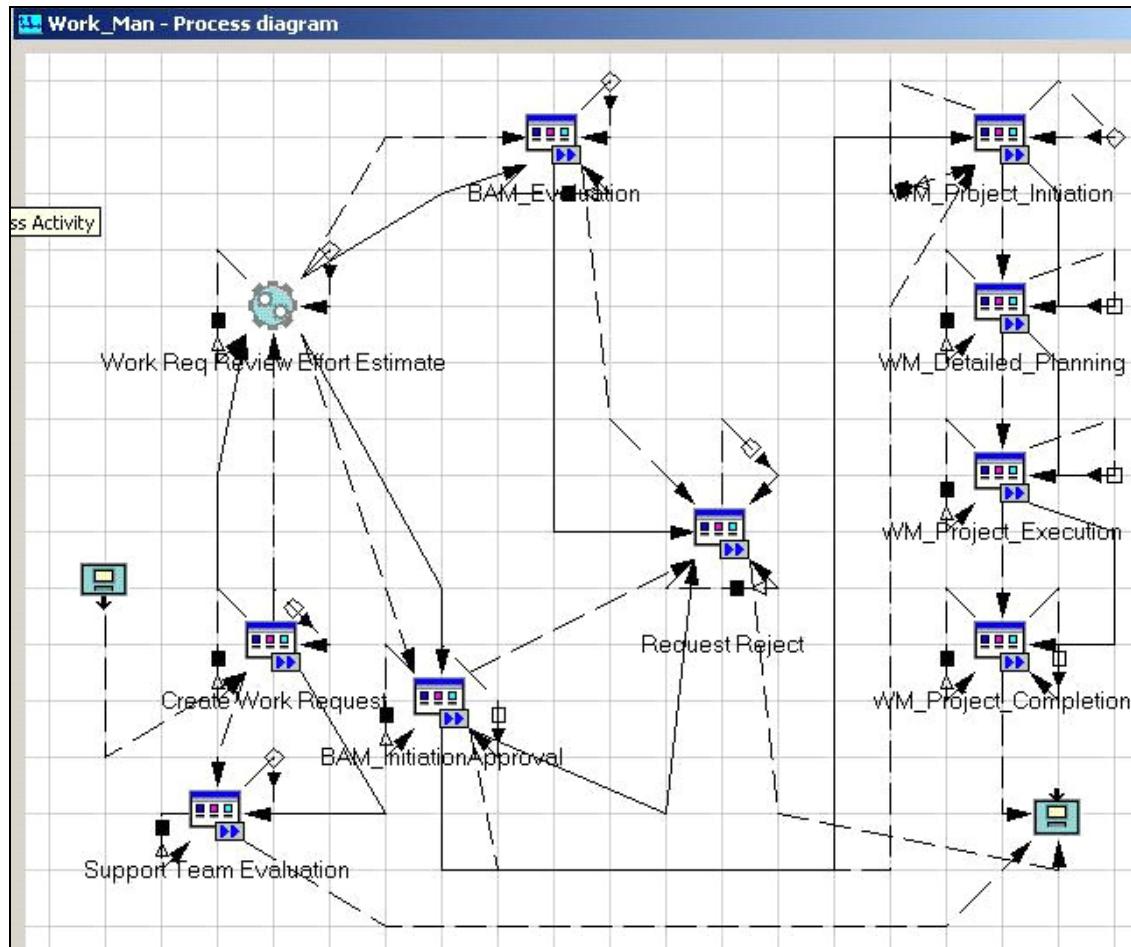


Figure 5-22 Work management template

5.4.4 Template for order processing

The Order Handling Process covers issuing and accepting orders. This includes sub-processes such as:

- ▶ Pre-order feasibility studies
- ▶ Credit verification and authorization
- ▶ Order issuing and configuration
- ▶ Order tracking and completion
- ▶ Customer satisfaction validation to measure and benchmark expectations

This template helps companies manage customer orders and configuration requests efficiently.

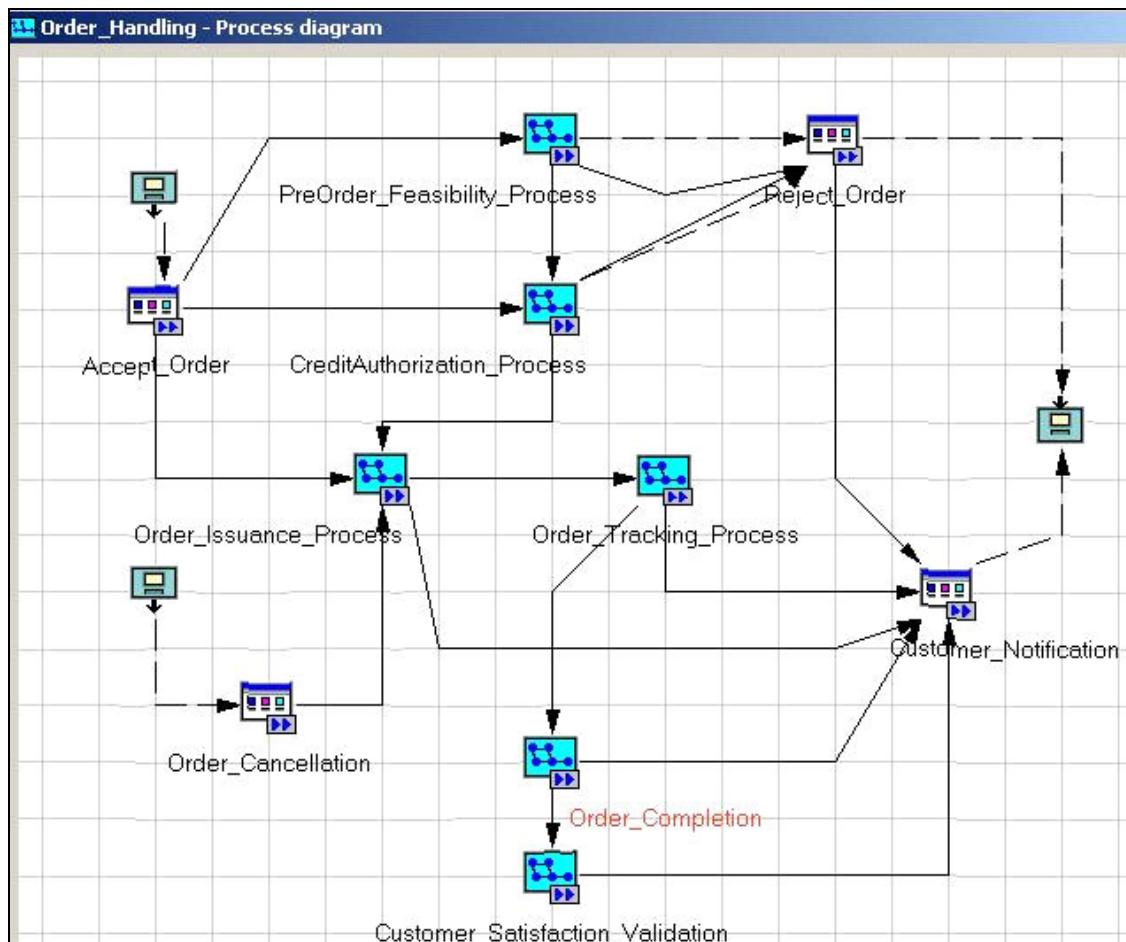


Figure 5-23 Order Handling Process

ePCM can be used to model any enterprise business process and deployed in a matter of days. Modeling business processes can provide solutions for other business verticals (e.g., in the Financial, Health and Pharmaceutical, Energy, and Retail industries). Portal version of ePCM enhances use of existing infrastructure and unified access to various business processes/applications. For example, the above scenario shows a unified access through WebSphere Portal to engineering data using applications like ePCM and ENOVIA 3d com Navigator. ePCM leverages WebSphere Portal functionality such as:

- ▶ User registration
- ▶ User management
- ▶ Authorization
- ▶ Authentication
- ▶ Single sign-on and security features enabling standard access to enterprise applications and content
- ▶ Personalization
- ▶ Integration with collaboration tools like Lotus Instant Messaging

5.4.5 Workflow federation

Business processes span multiple applications. Since any of these might possess workflows unique to their respective systems, there is a need for orchestrating these individual workflows to provide a seamless end-to-end solution. ePCM acts as an enterprise-wide federator of such workflows. Using the WebSphere Business Integration framework, ePCM federates other enterprise applications (e.g., ERP, PDM, and legacy applications).

Figure 5-11 on page 232 shows ePCM mapping against enterprise architecture specific to the Automotive and Aerospace Industries. ePCM can easily map to enterprise architecture in other industry verticals.

5.4.6 Other ePCM features

Object Manager

Since business objects usually have a definite lifecycle, they can be seen as “Evolving Business Objects.” ePCM has an Object Manager to represent basic business objects such as RFCs, ECRs, or ECOs as well as any custom-built business object. Objects can be built, visualized, and searched using the ePCM Object Manager—which also provides support for inheritance and dependencies—and attached to individual process instances to enhance information.

Workflow monitoring and e-reporting

ePCM audits processes and activities through its Process Monitoring and e-Reporting capability. This monitor helps users attain a global view of the process. The e-Reporting feature queries audit trails to report statistics such as Process Cycle Time, Start Time, End Time, Work Time, and Wait Time of processes and activities. The reporting tool also includes such features as comparison between two process instances, charting and graphical representation of various statistics, Quick Search, and preferred search capabilities.

Document Manager

The Document Manager module helps implement the paperless office. It can be used to connect documents and other related files to a process instance while supporting multiple formats and managing different versions of file attachments. It also tracks the history of each attachment. External content management systems can be used to manage these various document versions through stored parameters. The Viewer Center provides 2D/3D visualization as well as manipulation and annotation of standard CAD formats.

EAI framework

The EAI platform uses a hub-and-spoke architecture for IBM products such as WebSphere Interchange Server⁹⁴ and WebSphere MQ. This integration uses the connector framework provided by WebSphere Business Integration as well as custom-built connectors. ePCM connects to ENOVIA VPM⁹⁵ using the Galaxia VPM Connector, to LCA, and to ERP systems.

⁹⁴ Formerly CrossWorlds

⁹⁵ Virtual Product Model

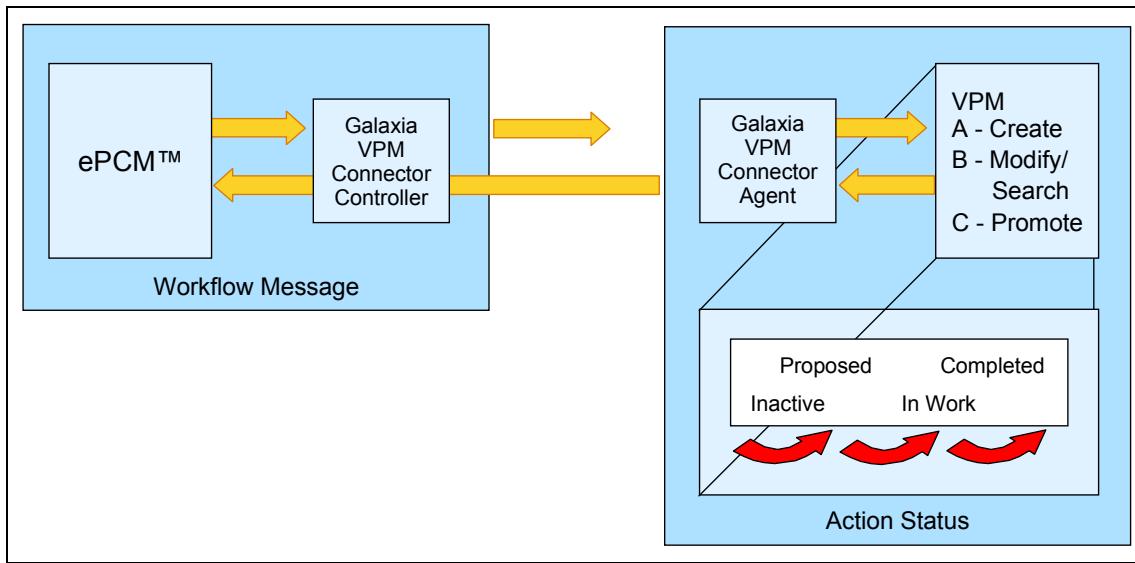


Figure 5-24 Galaxia VPM Connector

Note: In Figure 5-24, “VPM” refers to ENOVIA VPM.

The Galaxia VPM Connector Controller translates workflow messages sent from ePCM into ENOVIA VPM commands (for example, to create an Action object). The Controller Agent reads the ENOVIA VPM commands and executes them. On completion of the Action flow, the status of the Action Object is set to *completed*. The Connector Agent then translates the ENOVIA VPM message to its corresponding Workflow message.

The Connector supports creation, modify, search, and promote commands from ePCM.

Following are some of the advantages that can be realized by using the ePCM VPM Connector:

- ▶ Automatically updating legacy systems with engineering changes from your PDM systems
- ▶ Ensuring that builds are updated to the most current revision
- ▶ Preventing missed delivery commitments by enabling engineering and manufacturing departments to synchronize product data
- ▶ Reducing time to market.
- ▶ Receiving ECRs from the client's customer electronically and sharing information with other processes

Collaborative Suite

ePCM supports collaboration through work item notifications. ePCM has e-mail, chat, and whiteboarding capabilities and can be integrated with standard collaboration tools such as Lotus Team Workplace and Instant Messaging.

5.4.7 ePCM architecture

The modularity of the ePCM architecture and its compliance to J2EE makes it a more flexible, extendable, and adaptable Web solution.

This approach makes ePCM more portable and platform- and software-independent. ePCM allows the generation of object and process definitions without any additional programming effort. All software components (codes, programs, and user interfaces) are automatically generated, which significantly reduces the development, deployment, and maintenance lifecycle.

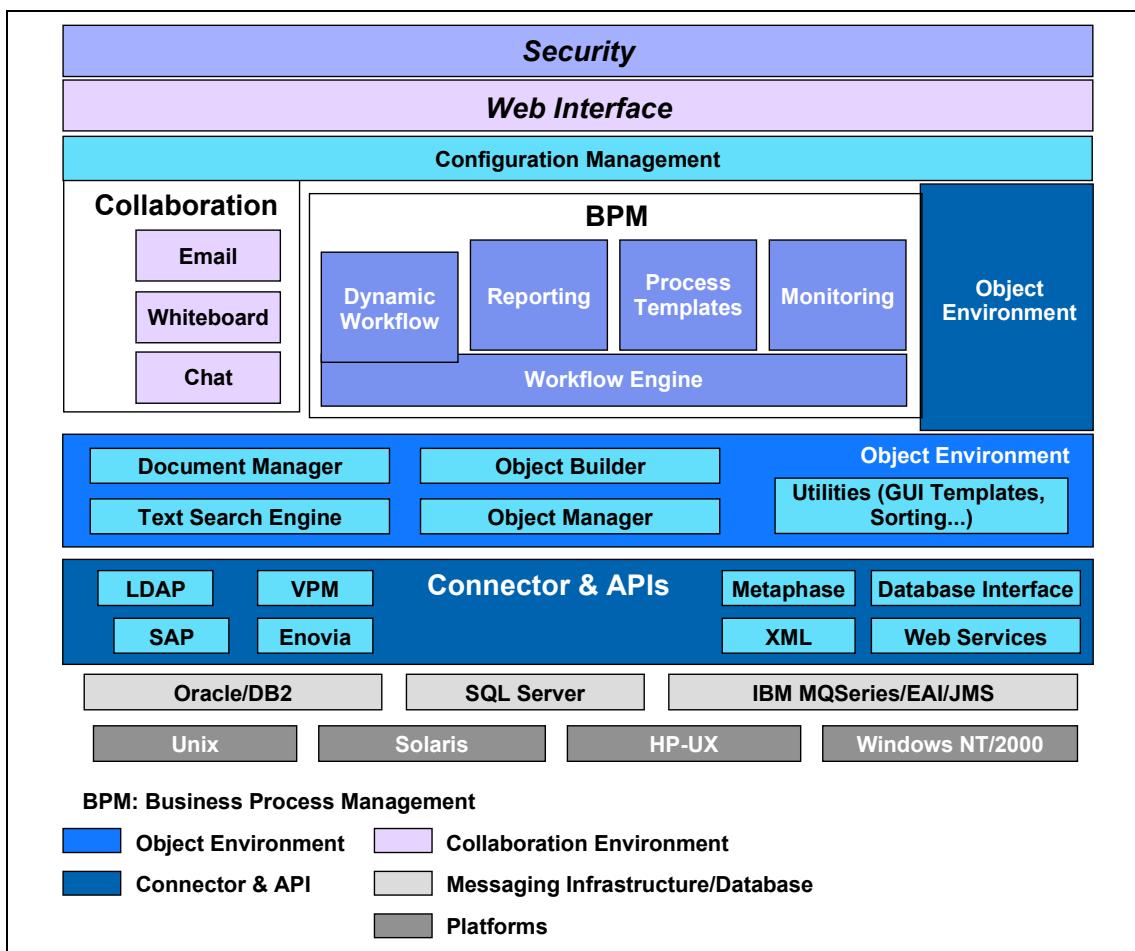


Figure 5-25 ePCM architecture⁹⁶

⁹⁶ Here "VPM" refers to ENOVIA VPM.

Table 5-9 ePCM modules

Modules	Details
BPM	<ul style="list-style-type: none"> ▶ Workflow ▶ Process Template ▶ Monitoring ▶ e-Reporting
Collaboration	<ul style="list-style-type: none"> ▶ e-Mail ▶ Chat ▶ Whiteboard
Configuration	Management of the access required to process templates and process instances
Object Environment	<ul style="list-style-type: none"> ▶ Object Builder ▶ Object Manager
Connectors and API	<ul style="list-style-type: none"> ▶ LCA connector ▶ Teamcenter Enterprise connector ▶ SAP connector ▶ ENOVIA VPM connector ▶ Web service ▶ XML ▶ LDAP connectivity
Document Manager	File attachment and version management, etc.
Web Client	Modifiable GUI Template

Key benefits of ePCM include:

- ▶ Federating enterprise-wide workflow and business processes
- ▶ Rapid design, optimization, and deployment of business processes
- ▶ Automation and management of mission-critical business processes
- ▶ Audit trails and reporting to enable global process visibility and monitoring
- ▶ Enterprise application integration capabilities for:
 - PDM
 - ERP
 - Legacy systems
- ▶ A Business Object Manager to manage various business objects
- ▶ Dynamic workflow to handle exceptions and unplanned tasks
- ▶ Dramatically lowering:
 - TCO⁹⁷
 - License costs

⁹⁷ Total Cost of Ownership

- Infrastructure costs
- Implementation costs
- Maintenance costs
- ▶ Ease of implementation (solution in production in six months or less)
- ▶ Enterprise collaboration capabilities, including:
 - e-Mail
 - Chat
 - Whiteboard
- ▶ A Web-based solution to involve the extended enterprise

5.5 Summary

Galaxia was able to leverage IBM technology and best practices to create a WebSphere Portal-based On Demand Workplace to improve process efficiency in the industrial sector. Incorporating ePCM, Galaxia architected a Web-based solution for the change management process, integrated with several key applications in the industry and extended the enterprise through the use of collaborative tools. The On Demand Workplace demo scenario brought out the key solution differentiators and benefits. The key benefits can be summarized under two different categories:

- ▶ **Access**—The architecture is portal based and hence has the advantage of single point of access. Federative Business Process flows encourage process sharing and application integration. Federative workflows can thus virtually access any enterprise data and present it in a coherent, standard format. This is one of the prime advantages for reducing slack and process cycle times in Collaborative Engineering domain. It enables access to right information at the right time for effective decision making. WebSphere Portal server being one of the basic infrastructure in this solution provides easy access to other existing IBM brand tools like Lotus Instant Messaging, thus spurring collaborative teamwork.
- ▶ **Business Integration**—The solution aims at a standard framework for enterprise-wide application integration. It leverages the WebSphere Business Integration Platform to provide connectivity to various enterprise applications. Process to process communication and effective choreography of various disparate application workflows leads to a totally integrated solution.

Although the solution showcased is aimed at solving issues in Collaborative Engineering for Automotive and Aerospace verticals. The framework being standards based can easily be extended to cover enterprise-wide business processes in other industry verticals.

Note: Click [here](#) to schedule a Galaxia demo.^a

a. www.galaxia.ca/epcm

Appendices

- ▶ Appendix A, “e-business on demand” on page 259

Explains how you can integrate heterogeneous systems and platforms using a roadmap that incorporates:

- Java and open standards
- Linux
- Web services
- Grid computing
- Autonomic computing
- Utility computing

Details key IBM products designed to get you there, including the following core recommended as your best starting point:

- WebSphere Studio, Application Server, and Business Integration Server
- DB2 Universal Database™ and Information Integrator
- Lotus Discovery Server™, Domino 6, and Lotus Instant Messaging
- Tivoli Identity Manager
- Rational Rose® and XDE™

- ▶ Appendix B, “Using WebSphere Portal” on page 383



A

e-business on demand

e-business on demand is IBM's position on how enterprises can best transform themselves to successfully face the challenges of an increasingly globalized economy. This implies a transformation to a fully integrated business across people, processes, and information—including suppliers and distributors, customers, and employees.

Definition: IBM defines an *on demand business* as an enterprise whose business processes—integrated end-to-end across the company and with key partners, suppliers and customers—can respond with speed to any customer demand, market opportunity or external threat.

The on demand era is now upon us—a time when the stakes are higher than ever, the perils significant, and the opportunities enormous—a time when everyone who makes your business run—customers, suppliers, partners, and shareholders—will expect everything you do to be delivered better, faster, and more cheaply.

This appendix is a collection of previously published materials detailing IBM's on demand vision.

A.1 e-business on demand: A developer's roadmap

Note: This article is by Alfredo Gutierrez—Manager, IBM developerWorks® Developer Skills Program (February 17, 2003). IBMers can find this article at [http://www-106.ibm.com/developerworks\(ibm/library/i-ebodov/.](http://www-106.ibm.com/developerworks(ibm/library/i-ebodov/.)

What is it all about, and what does it mean to you?

IBM has defined e-business on demand as an enterprise whose business processes—integrated end-to-end across the company and with key partners, suppliers, and customers—can respond with speed to any customer demand, market opportunity, or external threat. Are you wondering just what this means for you? Developers and IT professionals will be expected to build the technical infrastructure to support the integrated business processes of e-business on demand. To help you get up to speed, this article describes the e-business on demand environment and gives you the technology roadmap to get there.

It's all about return on investment. In the beginning, the Internet linked scientists in academia, government, and research. It evolved to provide e-mail and then the World Wide Web, which was good for communicating market messages but didn't have a lot of business value. Technology quickly evolved and enabled computing on the Internet, driving business processes. But this capability has come with a cost—it has required serious investment in technology. And when a business invests in information technology, it expects to derive benefits from its investment. Issues of cost reduction never go away.

So what's the problem?

There's a gap between what IT promises and what it delivers. You're a developer—you know that integrating disparate, heterogeneous systems and networks is complex. This complexity is the number one issue troubling CIOs today. Just trying to get technologies to work together eats up more than 40 percent of IT budgets. That means almost half the IT investment goes toward things that don't directly drive business value. Because it's complex, it can take months, maybe a year, before an IT investment delivers any value. Because it's complex, skills are in short supply, and it will get harder to hire the people to integrate, implement, and maintain technologies. Complexity costs.

And then there's utilization costs. Did you know that:

- ▶ Mainframes are idle 40% of the time.
- ▶ UNIX servers are idle 90% of the time.
- ▶ Most PCs are idle 95% of the time.

Of course, the industry grappled with cost of ownership and utilization long before the Internet and e-business introduced a new era of computing. Now we have the Web, but the promise of complete business integration efficiency still lies in the next generation of e-business technology infrastructure. And this is where you—the developer—come into play.

In this article, we'll take a tour of the e-business on demand environment and look at what businesses are demanding from a technology infrastructure. Then we'll talk about how you can integrate heterogeneous systems and platforms using a roadmap that incorporates these technology milestones:

- ▶ Java and open standards
- ▶ Linux
- ▶ Web services
- ▶ Grid computing
- ▶ Autonomic computing
- ▶ Utility computing

Just what is e-business on demand?

You are in an e-business on demand environment when your organization connects its core business systems to key constituencies using intranets, extranets, and the Web, allowing you to:

- ▶ Build and enhance business relationships through the thoughtful use of network-based technologies
- ▶ Leverage Internet technologies to transact and interact with customers, suppliers, partners, and employees to achieve and sustain a competitive advantage

Getting to e-business on demand is a natural progression that typically goes through these stages:

- ▶ **Access**—Enable transactions against core business systems using simple Web publishing and point solutions.
- ▶ **Enterprise integration**—Use the Web to integrate business processes across enterprises. Link internal and external systems, both across enterprises and beyond enterprise boundaries.
- ▶ **e-business on demand**—Use the Web to adapt dynamically to customer and market requirements. Change business models. Combine people, technologies, and processes in new ways.

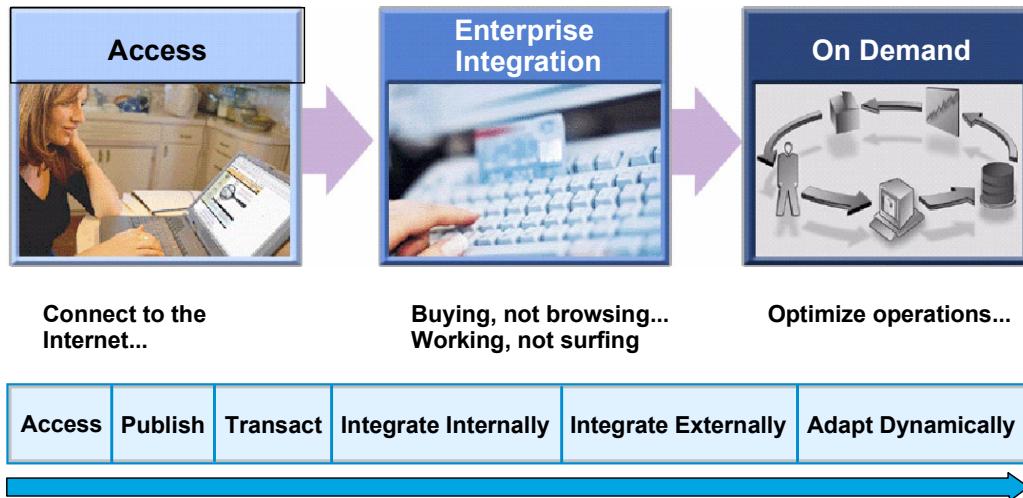


Figure A-1 A natural evolution

Phases of e-business on demand

e-business on demand evolves in phases. In each phase, the Internet transforms the business processes.

- ▶ **Access to digital information**—This phase is all about publishing content, most of it of the static “look-up” variety. Simple database queries allow us to check a bank account, look up airline flight information, or see where our overnight package is. It's pretty easy to get in the game here. All an enterprise needs is a home page. All an individual needs is a browser.
- ▶ **Real transactions, real e-business**—Don't just look at your bank account—move some money. Don't just check a flight departure time—book your seat. Trade a stock, buy a book, apply for a loan, renew your driver's license, take a college course. Doing this requires more than a Web site—this requires behind-the-scenes integration of technologies and business processes.
- ▶ **The advanced stage of e-business**—In a fluid system of customers, suppliers, partners, and employees, the Internet is the primary way to communicate, transact, and connect. Business processes shift from manual to automated. A relationship could last only as long as a single transaction. The environment is real time computing. You form networked communities so that organizations can:
 - Create new products and services faster
 - Reach new customers and economically add new relationships
 - Dynamically change existing relationships
 - Simultaneously engage in multiple e-business models

- Improve access to information by constituents involved in these relationships

What's the return on investment for an enterprise that progresses through these phases? An increased share of customer spending, a better return on assets, new revenue opportunities, and better shareholder return.

Every industry segment has the common requirement to integrate end-to-end so that products, services (private sector and government), invoices, images, decisions, and answers are all available on demand. Who will have the competitive advantage? The enterprises that get there first, and the software and services providers that know how to make it happen.

Any globally connected enterprise must have the ability to handle whatever comes its way, such as changes in customer preferences or competitive actions, fluctuations in capital markets, labor situations, natural disasters, or political unrest. An advanced e-business must be able to respond to the unpredictable and the unforeseen and never question the ability of the infrastructure to deliver.

Which phase is your enterprise in? About 75% of all businesses are in Phase 1. Another 20% are in Phase 2 and are already reaping the benefits. These businesses have identified key processes—from procurement of raw material to the manufacturing floor, to shipping and distribution, to customer relationship management—and they're applying networked technologies to transform them. These enterprises have determined that transforming a standalone process is a beginning, not an end.

So what kind of computing environment is required for an on demand business? What does “on demand” mean for the way a business buys and manages its computing technology? Before tackling these questions, let's look at the four business characteristics of an on demand business:

Business characteristics of an on demand environment

An on demand business has these characteristics:

- ▶ **Responsive**—Able to sense changes in the environment and to respond dynamically to unpredictable fluctuations in supply or demand, emerging customer, partner, supplier and employee needs, or unexpected moves by the competition.
- ▶ **Variable**—Able to adapt cost structures and business processes flexibly, to reduce risk, and to drive business performance at higher levels of productivity, cost control, capital efficiency, and financial predictability.
- ▶ **Focused**—Committed to concentrating on core competencies and differentiating tasks and assets; able to use tightly integrated strategic

partners to manage tasks ranging from manufacturing, logistics, and fulfillment to human resources and financial operations.

- **Resilient**—Prepared for changes and threats like computer viruses, earthquakes, or sudden spikes in demand.

A business with these attributes requires technology that can support it, but that's not the computing environment that's operating today. No, today's environment is heterogeneous, widely distributed, vertically isolated, and generally more complex than businesses would like. The same IT that's essential to a business's ability to create strategic advantage, is also a major obstacle to becoming the kind of fluid, responsive, dynamic business that's been talked about for years.

Technical characteristics of an on demand environment

Based on the above business characteristics of an on demand business, what kind of technology environment is needed to support it?

It's integrated

Beyond just transformation department by department, business processes and applications need to integrate horizontally to link data, legacy systems, and custom applications, and this integration requires new levels of data integrity and transaction processing. You will need an infrastructure built on Web services, new development tools, and open standards. Applications that had previously integrated vertically—with an operating system and standalone processor—must now integrate horizontally, application to application. You will write applications to the middleware layer, not to the operating system. Applications are being decoupled from the underlying infrastructure.

It's open

There's no other way. In a networked world, you have to do more than simply integrate everything inside your enterprise. You have to connect your enterprise with other enterprises, other business processes, other applications, and billions of pervasive computing devices. You can't just rip and replace your existing data, applications, and transaction systems to make them homogenous with those of your business partners. Open specifications and industry standards are the only realistic way that all of this can connect.

It's virtualized

Every organization that has made capital outlays for technology is sitting on enormous, unused computing capacity. Server consolidation and capacity on demand offerings begin to address the issue of under-utilization. But now there's an opportunity to virtualize the entire data center with an emerging technology called grid computing. Grid computing allows distributed computing resources to be shared and managed as if they were one, large, virtual computer. Initially most grids are being built in government laboratories and universities. From

there, they will be implemented inside companies. These “intragrids” will allow enterprises to increase the utilization of their own computing assets.

It's autonomic

The upward spiral of complexity will soon exceed the ability of skilled technical, human resources to keep up with it. The solution? Computing systems that take on more of the management themselves. In the same way that the human autonomic nervous system manages basic functions like respiration, autonomic systems will self-manage, self-protect, balance workloads, install device drivers, and upgrade software. This requires innovation, real science at both the component and overall system level. In 2001, IBM's technical community articulated the capabilities and requirements for this kind of autonomic infrastructure. IBM Research defined the kinds of innovation required at the component, product, and systems levels. The industry has been invited to join us in the mastery of this computing challenge.

Utility computing

We've been discussing transformation on two levels:

- ▶ Transforming business processes
- ▶ Transforming the underlying technical infrastructure

Together, these transformations bring yet another transformation that may prove to be the most exciting aspect of the movement to e-business on demand: the way a business buys and manages its computing power.

Until now, your choices were limited. You needed more computing power? You bought more computers, or you outsourced. Now, on demand business requires a fundamental change in the way we access, pay for, and manage all the assets of the IT industry.

The answer is utility computing. You virtualize the data center and build internal computing utilities that drive up utilization and value delivered to the business. When this kind of virtualization moves to the Internet, you'll be able to tap into external utilities, get computing from service providers, and pay only for what you use.

Technology roadmap

If adaptive, dynamic integration is the goal, and utility computing is how you're going to implement it, where do you start? Focus on the key technologies: open standards and Java, Linux, Web services, grid computing, and autonomic computing.

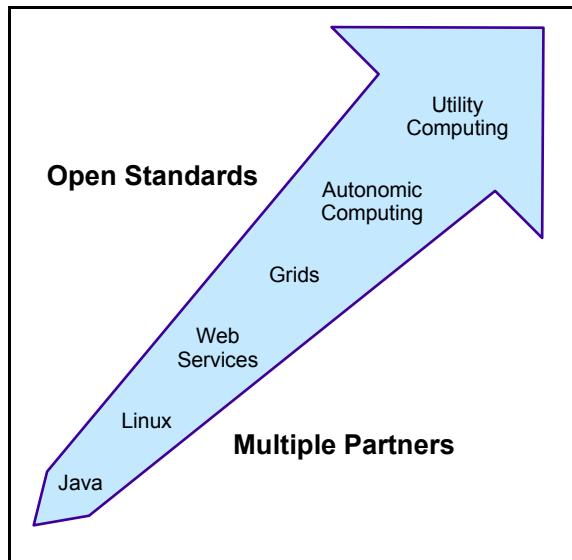


Figure A-2 Landmarks on the technology roadmap

Open standards and Java

You don't want to rip and replace. Instead, you want to link together your disparate, distributed, heterogeneous systems, and you can do this using open standards. You can use open standards to ease your integration burden and tie new products and technologies into your existing infrastructure easily and at low cost.

In your on demand environment, you must be able to add or reduce capacity quickly as the business requires. In today's heterogeneous environment, open standards with common interfaces let you choose function from a variety of vendors and snap it into the infrastructure. The IT industry is evolving into a culture of open standards.

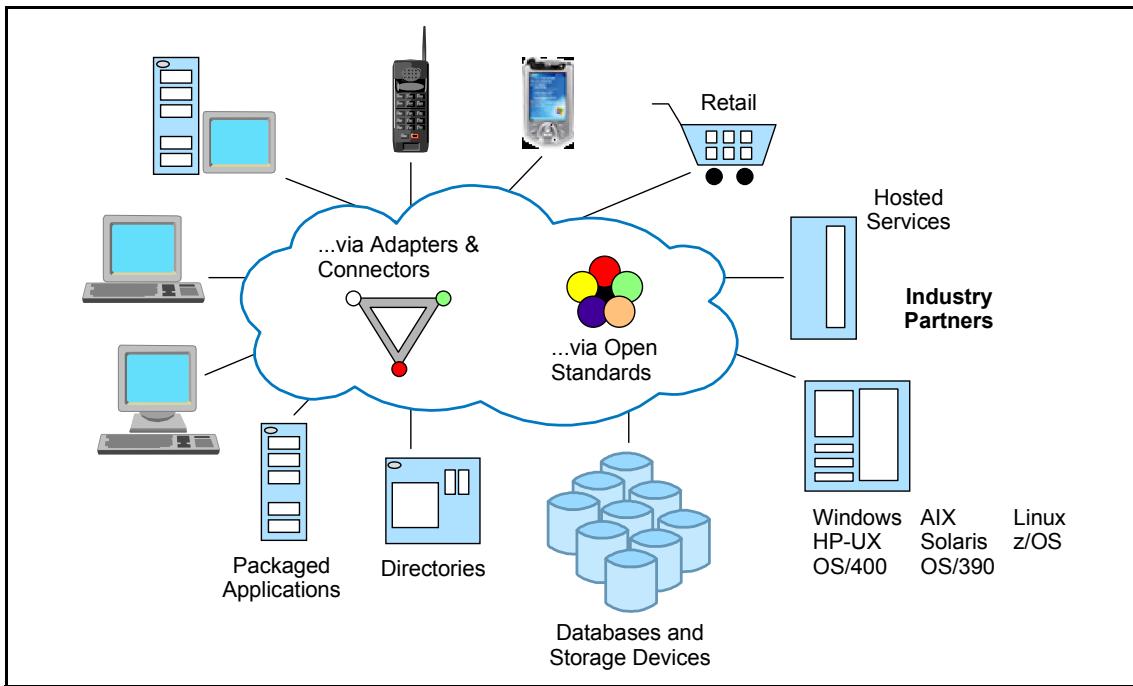


Figure A-3 Open standards speed integration

Linux

Why choose Linux? First, it's reliable, scalable, and secure. It's an enterprise-quality operating system, and you can trust Linux with your enterprise applications because it's a stable and mature base.

Second, Linux is about the lowest-cost alternative on the market, and that's a major factor in its growing appeal to business and government. It's easy to migrate your code from UNIX to Linux, and there's a sizeable pool of developers with Linux skills.

Plus, the applications you write for Linux can run on any platform. You can choose the server that's right for the application instead of the application that's right for the server. This flexibility frees you to innovate and create new applications, rather than spending your time rewriting old ones.

Finally, Linux provides an open, standards-based application platform, especially when combined with J2EE and Web services. Open standards make sense because they allow you to choose the best from among a wide variety of products and integrate them easily, rather than having to settle for whatever fits into a proprietary environment. This is a new model for the industry—companies

compete to produce the best products on this open standards base, instead of competing to establish a proprietary hold in a particular industrial segment.

Linux and the IBM eServer™ products: For IBM, Linux is consistent with our principles of open standards-based computing and platform independence. We have solutions for Linux across all our hardware platforms, and our middleware provides extensive support for Linux across the entire eServer set of platforms.

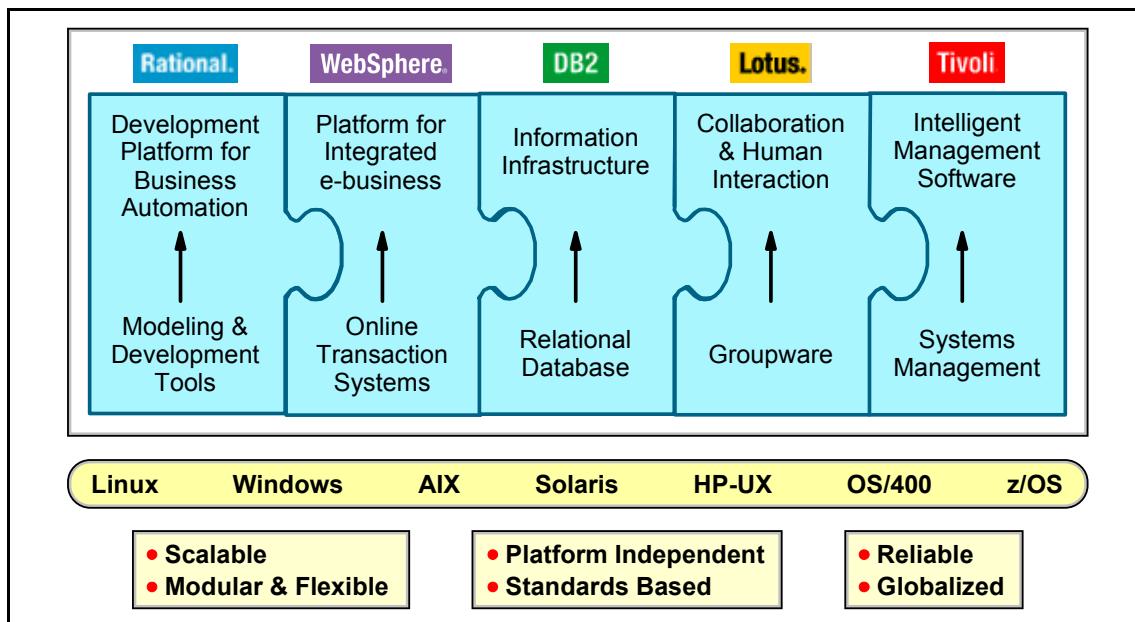


Figure A-4 Comprehensive Linux support in IBM software

Web services

Businesses have been searching for a technology solution to enable their infrastructures to be as flexible as their highly fluid business models. They've found an answer in Web services architecture, a set of industry standard methods that enable simplified programmatic connections between applications. Web services focuses on simple, Internet-based standards to address heterogeneous distributed computing.

Applications designed within a Web services architecture can seek each other out, integrate and execute transactions, all in an automated fashion. The advantages for a business are clear. A manufacturer could automatically connect with the supplier that best meets its cost and technical demands, while that

supplier in turn could connect automatically with manufacturers that have similar needs.

Web services and IBM software: IBM has enabled its software portfolio (from WebSphere to DB2, Lotus, and Tivoli) to support the development, deployment, storage, and management of Web services solutions.

Grid computing

A *grid* is a collection of distributed computing resources available over a network that appear to an end user or application as one large virtual computing system. A grid can span locations, organizations, machine architectures, and software boundaries to provide unlimited power, collaboration, and information access to everyone connected to the grid. The effect of grid computing is to make network computing more like a utility. You deliver computing power to where you need it only when you need it; you pay for what you use, when you use it.

Like the Internet, grids started in the scientific community but are now being deployed by business enterprises. Recently, IBM and Globus collaborated to combine open grid protocols and Web services standards, producing the Open Grid Services Architecture (OGSA).

Autonomic computing

Autonomic computing systems are:

- ▶ **Self-configuring**—Able to adapt to dynamically changing environments
- ▶ **Self-healing**—Able to discover, diagnose, and act to prevent disruptions
- ▶ **Self-optimizing**—Able to tune resources and balance workloads to maximize use of IT resources
- ▶ **Self-protecting**—Able to anticipate, detect, identify, and protect against attacks

When you implement autonomic systems, you are free to focus on more strategic and higher-level issues. For the business, the core benefits of autonomic computing are improved resiliency, ability to deploy new capabilities more rapidly, and increased return from IT investments.

Autonomic computing capabilities play a critical role in the development of grid computing. Grids can become the most complex computing environments available. Autonomic computing will allow grids to be easily managed and to ensure that they deliver the levels and quality of service demanded by businesses.

Utility computing

The concept of e-sourcing is simple. It's Information Technology as a utility. Think of electricity, or telephone service. You don't need a generator to get electricity to your home or office. You just plug into the electrical grid, use what you need, and pay for what you use. An enterprise will be able to focus maximum resources on its core business, on what differentiates them from their competition.

Where's the data center? It's in the enterprise. It's outside the enterprise. Or it's shared between them. It doesn't matter where, because the infrastructure becomes a pool of resources available on demand. Like electricity, computing becomes an on demand, pay-as-you-go service—a utility so reliable that a business can take it more or less for granted.

Once grids enable computing resources to be shared globally and managed autonomically from end-to-end, an enterprise's infrastructure becomes incredibly flexible. It can deliver computing power itself, but also switch to a supplier for peak power, since they both operate on the same standards. The infrastructure is a pool of virtual resources a customer can call on as needed.

In an on demand future, a company might respond to increasing demand for computing resources first with spillover services from their own IT utility and, if that isn't enough, with capacity from a service provider, purchasing only the extra capacity needed at the moment. Because provider and customer share common, open protocols and use grid and autonomic technologies, all sorts of services can be provided dynamically in the smallest, most economical blocks. Imagine paying for increments of computing power as used instead of for large volumes, long term. The potential impact on budgets and the bottom line is huge.

The concept of IT delivered as a utility isn't new; but only now are technological advances making it possible:

- ▶ Reduced cost of bandwidth, which enabled the creation of new data services and high-speed network delivery of a variety of services to a broader range of customers
- ▶ Distributed content and application architecture deployments that shift delivery to the edge of the network
- ▶ Server and storage virtualization, which enables a new level of shared infrastructures with the potential of reducing customer costs

Over time, as the focus turns more toward the application and the business rather than the infrastructure, the question will become: "Do I really need all that infrastructure?" To put it another way, "Do I want to generate my own electricity, with all the investment in people and equipment that entails? Or am I comfortable leaving the generation of electricity to the experts and IT to the IT professionals?"

Conclusion

Add them all together: You have applications and systems integrated using open standards. Plus Web services that provide definitions, discovery, and access to self-managing, autonomic IT resources on a grid. And what do you get? You get computing resources capable of being shared globally and managed end-to-end. You get an infrastructure that's incredibly flexible and that will allow new capabilities to be deployed with relative ease. This is the technical environment on which an on demand enterprise depends.

With e-business on demand, information technology will change. IT will be delivered as a utility. Utility computing will offer fundamental advantages over the current model. You'll get there in stages, and you can start today by following the technology milestones we've outlined: Java and open standards, Linux, Web services, grid computing, and autonomic computing.

A.2 Resource center: e-business on demand

Note: The material in this section came from the IBM developerWorks e-business on demand Resource Center Web site. IBMers can find this article at <http://www-106.ibm.com/developerworks/ondemand/>.

If you want to radically improve the efficiency of your business, you need tools and techniques that can help you integrate your business processes with your partners, suppliers, and customers. You need to respond to change with flexibility and speed. You need to evolve into an e-business on demand environment.

More than “e-sourcing,” or utility-like computing, e-business on demand is a much broader idea about the transformation of both business and technology.

This transformation is based on open standards—so different systems can work together and link with devices and applications across organizational and geographic boundaries. (Using open standards assures that adoption of on demand computing will not lead to your business becoming wed to IBM, or tied too closely to any single vendor.)

From the technology standpoint (the primary focus for developers), an on demand environment has three essential capabilities:

- ▶ **Integration**—Systems are seamlessly linked across the enterprise and across its entire range of customers, partners, and suppliers.
- ▶ **Virtualization**—To make the best use of technology resources and minimize complexity for users, it uses grids to make the collective power of computing resources in the grid available to anyone in the grid who needs them.

- ▶ **Automation**—It has self-healing, autonomic capabilities—so it can respond automatically and work around problems, security threats, and system failures.

A.2.1 Open standards

In a networked world, you have to do more than simply integrate everything inside your enterprise. You have to connect your enterprise with other enterprises, other business processes, other applications, and billions of pervasive computing devices. You can't just rip and replace your existing data, applications, and transaction systems to make them homogenous with those of your business partners. Open standards allow all technologies to connect and integrate, and allow IT to become more modular. Linux and Java technologies brought open standards to the enterprise; today XML and Web services let you share information and applications across business lines.

Technologies

- ▶ **Java technology** began life as a way to add interactivity to Web pages through client-run applets and applications, but the most popular current use is in server-based J2EE systems. The developerWorks Java zone has an extensive collection of articles, tutorials, tools, and downloads for Java developers of all skill levels.
- ▶ **XML**, or Extensible Markup Language, is a markup language that you can use to create your own tags. It was created by the World Wide Web Consortium (W3C) to overcome the limitations of HTML, the Hypertext Markup Language that is the basis for all Web pages. Like HTML, XML is based on SGML—Standard Generalized Markup Language. XML was designed with the Web in mind.
- ▶ **Linux** is the fastest growing server operating system in the world, thanks to its powerful functionality, rock-solid stability, and open source foundation. Apps developed on Linux—from Web and e-mail servers to departmental and enterprise vertical applications—are reliable, portable, and cost efficient.
- ▶ Businesses have been searching for a technology solution to enable their infrastructures to be as flexible as their highly fluid business models. They've found an answer in **Web services** architecture, a set of industry standard methods that enable simplified programmatic connections between applications. Web services focuses on simple, Internet-based standards to address heterogeneous distributed computing.

Products

We recommend that you start with the following products:

- ▶ **IBM WebSphere Studio** is an open comprehensive development environment for building, testing, and deploying dynamic on demand e-business applications. Founded on open technologies and built on Eclipse, WebSphere Studio provides a flexible, portal-like integration of multi-language, multi-platform, and multi-device application development tools that maximize your productivity, increase ROI and improve overall time to value.
- ▶ **IBM WebSphere Application Server** is a high-performance and extremely scalable transaction engine for dynamic e-business applications. The Open Services Infrastructure allows companies to deploy a core operating environment that works as a reliable foundation capable of handling high volume secure transactions and Web services. WebSphere continues the evolution to a single Web services-enabled, Java 2 Enterprise Edition (J2EE) application server and development environment that addresses the essential elements needed for an on demand operating environment.
- ▶ You can use **WebSphere Business Integration Server V4.2** to quickly integrate new or existing applications or systems on diverse platforms, create and rapidly deploy new business processes, or solve a variety of business integration needs.
- ▶ **Rational XDE** provides a frictionless design and development environment. At the core of this eXtended Development Experience is Rational XDE Professional and Rational XDE Modeler, visual design and development tools that are tightly integrated with WebSphere, giving you a single user experience.

A.2.2 Integration

Integration is the heart of e-business on demand. Your business becomes more powerful when you integrate horizontally, connecting with the vast amounts of data, legacy systems, and custom business applications inside and outside your business. Integration gives you real time transaction processing, data mining, and decision support systems to bring your business up to Web speed.

Business integration

At the core of e-business on demand is business integration. Transforming into an on demand business requires building a dynamic infrastructure based on tightly integrated, streamlined critical business processes. Processes efficiently linked across your company and with those of key trading partners, suppliers and customers. Integrated business processes that give you flexibility, the ability to

respond immediately to almost any customer demand, market opportunity, or external threat.

To gain this flexibility, a well-thought-out integration strategy based on a robust platform is key. A platform for automating and managing value chain processes both inside and outside the firewall. To slash cycle times and costs. To speed time to market. To gain business agility in the face of competitive pressures.

Companies evolving into e-businesses on demand have made WebSphere Business Integration the cornerstone of their integration strategy. WebSphere features a solid integration foundation with the comprehensive e-business capabilities you need in an on demand era.

These five capabilities include:

- ▶ **Model**—Design, simulate and plan business processes.
- ▶ **Integrate**—Link people, processes, applications, systems and data.
- ▶ **Connect**—Extend processes to your customers and partners.
- ▶ **Monitor**—Control and track business processes.
- ▶ **Manage**—Review, analyze and improve processes and performance.

From enabling you to quickly redesign existing or deploy new processes, to easing the tracking and managing business events, a common thread runs across these five capabilities: real business results for both line of business management and IT.

You have the choice: deploy these capabilities separately or together to meet your business needs. Regardless, you have the leading platform, WebSphere Business Integration, for e-business on demand. Built with award-winning technology you can rely on to transform to an on demand business.

Data integration

DB2 Universal Database, WebSphere MQ, and WebSphere Studio Application Developer are three popular IBM e-business software products. Many customers have used one, two, or even three of these products in a single deployment. These products can function very well on their own, but are very useful for data integration when used together. This section explains how data integration can be achieved with each of these products using XML as a data interchange medium.

- ▶ **IBM DB2 Universal Database** is a stable, reliable, and scalable database that will allow us to store the data used in our business application.
DB2 Universal Database (DB2 UDB) Version 7 provides a scalable, Web-ready database that delivers the performance, scalability, reliability, and availability needed for the most demanding e-commerce, CRM, ERP, and Business Intelligence applications. DB2 UDB is a relational database for

today's heterogeneous computing environments and open standards that is able to access and integrate multiple data types from multiple geographically separated sources on different platforms.

- ▶ **IBM DB2 XML Extender** is an end-to-end solution for storing and retrieving XML documents.

DB2 XML Extender provides new data types to store XML documents in DB2 databases and new functions to work with these structured documents.

Managed by DB2, these documents are stored as character data or external files. Retrieval functions enable you to retrieve complete documents or individual elements.

- ▶ **IBM WebSphere Studio Application Developer** allows the professional application developer to quickly and easily build, test, integrate and deploy Java and J2EE applications.

The IBM WebSphere MQ product provides application programming services that enable application programs to communicate with each other using messages and queues. This form of communication is referred to as asynchronous messaging. It provides assured, once-only delivery of messages. Using WebSphere MQ means that you can separate application programs, so that the program sending a message can continue processing without having to wait for a reply from the receiver. If the receiver, or the communication channel to it, is temporarily unavailable, the message can be forwarded at a later time. WebSphere MQ also provides mechanisms for generating acknowledgements of messages received.

- ▶ The **Extensible Markup Language (XML)** is a key technology that facilitates information exchange and e-business transactions.

While these are useful in their own right, integrating them can be even more powerful. Business data is often a central component of many integration scenarios, hence the need for DB2. That data will more than likely need to be passed to other systems within the enterprise or to other organizations—XML fits the bill as a pervasive and useful transport medium for this data. The DB2 XML Extender provides the tools in order to easily interface these technologies.

WebSphere MQ is a very useful application in order to transport the business data via messaging. And lastly, WebSphere Studio Application Developer has all the right tools necessary to implement the integration, from database to XML support. In addition, if your integration scenario requires HTML pages, servlets, or Enterprise Java Beans, WebSphere Studio Application Developer is an excellent tool for writing J2EE applications, with an included WebSphere Application Single Server test environment.

Integration management

Note: The following is from “Tivoli Identity Manager 4.4 Logical Component Structure,” redbook tip (TIPS0045), published in April 2003.

The logical component design of IBM Tivoli Identity Manager may be separated into three layers of responsibility, which are depicted in Figure A-5. They are:

- ▶ The Web User Interface layer
- ▶ The Application layer
- ▶ The Service layer
- ▶ LDAP directory
- ▶ Database
- ▶ Resource connectivity

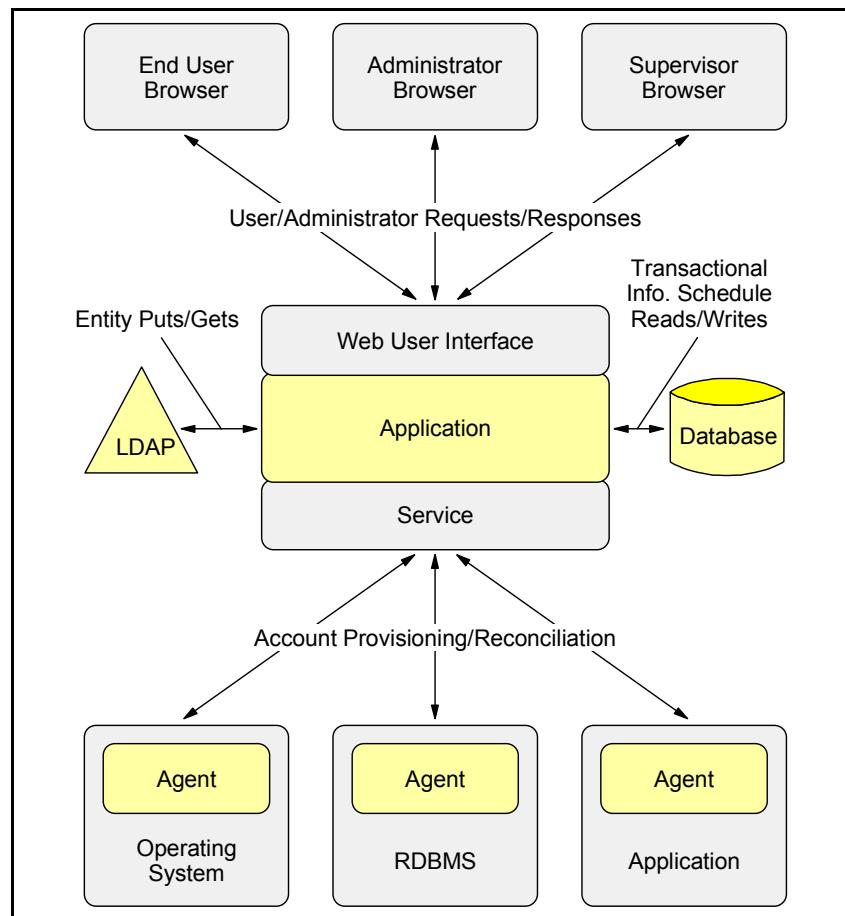


Figure A-5 Tivoli Identity Manager logical component architecture

Web User Interface layer

The Web User Interface module is a set of combined subprocesses that provide content to a user's browser and initiate applets (run both on the client and the server), such as the Workflow Design and the Form Creation. The Web User Interface is the interconnecting layer between the user's browser and the identity management application layer.

In Figure A-5 on page 276, there are three types of user interaction points: The end user, supervisor, and administrator. These types are merely conceptual, as IBM Tivoli Identity Manager allows the customer to define as many different types of users with different permissions as they'd like.

However, in this figure, it is important to note that the system is built with a general concept of the capabilities of the system users. For example, it is assumed that the administrator needs advanced capabilities and requires a more advanced user interface, possibly requiring a thicker client (applet). It is assumed that the supervisor needs slightly less advanced capabilities but may still require concepts like an organizational chart. Since the number of supervisors in an enterprise may be great, a thick client is not practical. Lastly, there are no assumptions made for the end user. The interface presented to the end user must be a thin client with very basic and intuitive capabilities.

The Web User Interface subsystem contains all modules necessary to provide a Web-based front end to the applications of the Applications subsystem, as shown in Figure A-6.

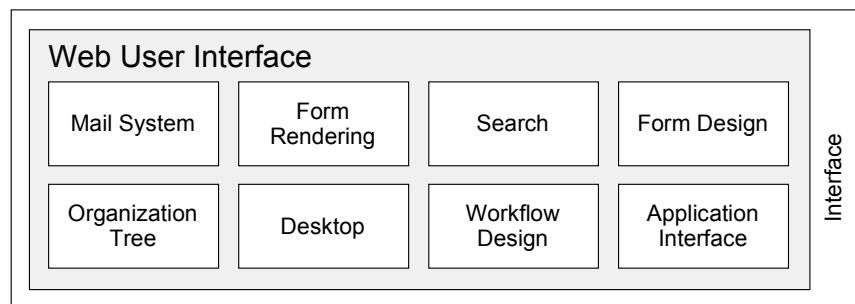


Figure A-6 Tivoli Identity Manager Web User Interface subsystem

Application layer

The core of the IBM Tivoli Identity Manager system is the Application Layer. Residing on an application server, the application layer provides the management functionality of all other process objects.

The Web User Interface subsystem contains all modules in the Platform subsystem that provide provisioning specific capabilities, such as identity

management, account management, and policy management. Each application makes use of the core services in the Platform subsystem to achieve its goals. It is the Applications module that provides the external interface to the provisioning platform. Below is a brief description of each module.

The Application Interface module consists of all application specific user interface components. For example, the interface required to create a provisioning policy or an account is organized in this module. This module makes use of other modules in the Web User Interface subsystem, such as the Form Rendering and Search modules. Figure A-7 shows the modules of the application layer.

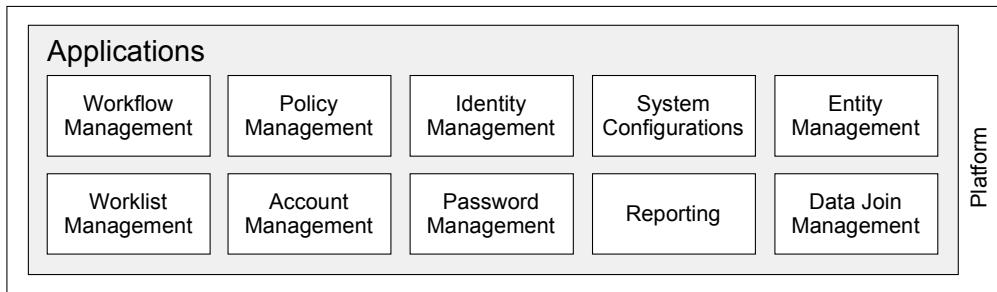


Figure A-7 *Tivoli Identity Manager Application Interface module*

Service layer

If the IBM Tivoli Identity Manager server is the application of complex rules that have been developed, then the applications server is the engine that runs those rules or objects. It is communicating not only to the user facing Web server, but also to the agents residing on the managed services and to directories for storage of information.

The Core Services subsystem contains all modules in the Platform subsystem that provide general services that can be used within the context of provisioning, such as authentication, authorization, workflow, and policy enforcement. These services often make use of other services to achieve their goals. Figure A-8 on page 279 shows the involved components.

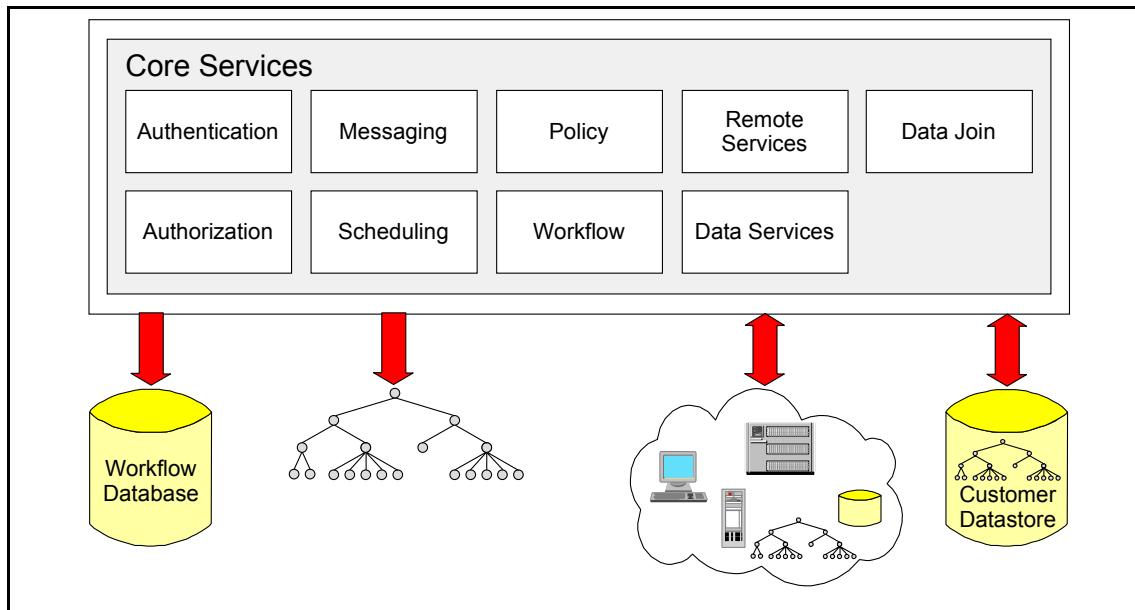


Figure A-8 Tivoli Identity Manager Core Services subsystem

LDAP directory

The IBM Tivoli Identity Manager system uses an LDAPv3 directory server as its primary repository for storing the current state of the enterprise it is managing. This state information includes the identities, accounts, roles, organization chart, policies, and workflow designs.

Database

A relational database is used to store all transactional and schedule information. Typically, this information is temporary for the currently executing transactions, but there is also historical information that is stored indefinitely to provide an audit trail of all transactions that the system has executed.

Resource connectivity

The back-end resources that are being provisioned by IBM Tivoli Identity Manager are generally very diverse in their capabilities and interfaces. The IBM Tivoli Identity Manager system itself provides an extensible framework for adapting to these differences to communicate directly with the resource. For a more distributed computing alternative, a built-in capability to communicate with a remote agent is provided. The agents typically use an XML-based protocol (DSML—Directory Services Markup Language) as a communications mechanism.

Directory Services Markup Language connectivity

DSML provides a method for processing structured directory based information as an XML document. DSML is a simple XML schema definition that enables directories to publish profile information in the form of an XML document so that it can be easily shared via IP protocols, such as HTTPS, which is shown in Figure A-9.

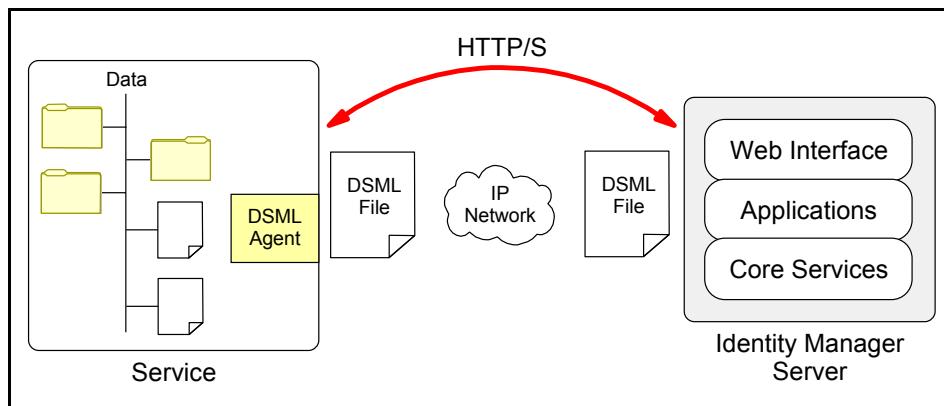


Figure A-9 Tivoli Identity Manager DSML connectivity

Transactions from the IBM Tivoli Identity Manager server are sent securely via HTTPS to the service agent and then processed by the agent.

For example, if a service has just been connected to the IBM Tivoli Identity Manager server, the accounts that already exist on the server may be reconciled or pulled back in order to import the users' details into the IBM Tivoli Identity Manager LDAP directory. If a password change or a provisioning of a new user occurs, the information is transferred to and then processed by the agent. The agent deposits the new information within the application or operating system that is managed.

IBM Directory Integrator connectivity

Due to the nature of support issues for every adaptor and the development costs for new adaptors, it is becoming easier to use IBM Directory Integrator as the data bus that transports information and data to the services. Using IBM Tivoli Identity Manager's JNDI API, it is possible to connect to just about any service, application, or endpoint, as shown in Figure A-10 on page 281.

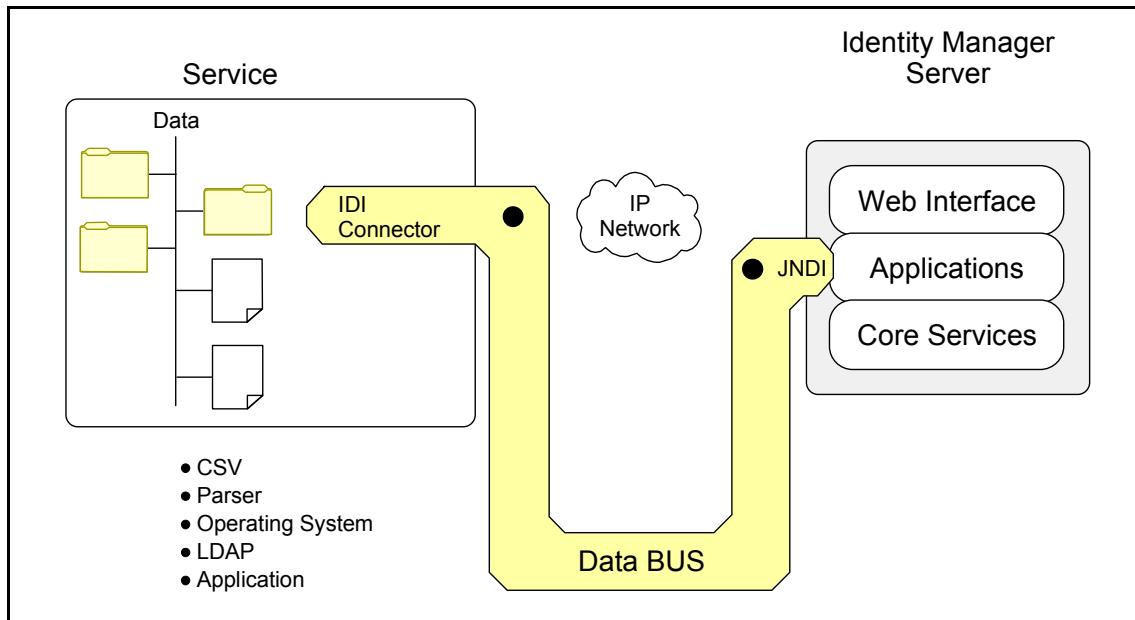


Figure A-10 Tivoli Identity Manager IBM Directory Integrator connectivity

By using IDI and IBM Tivoli Identity Manager, communications to resources such as a CSV file or a Microsoft Excel spreadsheet containing contractor employment details may be possible through a parser connector. Tivoli Identity Manager would see this as a service that it manages and that can be incorporated into its workflow.

Integration through Web services

Note: The following material comes from “Understanding Web Services” by Arthur Ryman (Senior Technical Staff Member, WebSphere Studio Development, IBM Toronto Lab), published in July 2003. IBMers can find this at http://www7b.software.ibm.com/wsdd/library/techarticles/0307_ryman/ryman.html.

In the beginning, there was the *Document Web*. The Document Web was powered by Web servers that spoke Hypertext Transfer Protocol (HTTP) and delivered information marked up in Hypertext Markup Language (HTML) for display on Web browsers. The Document Web was created to serve the needs of scientists who needed to share research papers. Businesses saw that the Document Web was a good thing, and many businesses created Web sites to advertise their products and services.

Then businesses wanted to do transactions with their customers, so the *Application Web* was created. The Application Web was powered by Web application servers, such as IBM WebSphere Application Server, which dynamically generated HTML documents from server-side business logic written in programming languages such as Java. Web application servers acquired distributed programming capabilities, so that they could scale over multiple machines to handle large transaction rates. Web application servers also learned to speak Wireless Application Protocol (WAP) and deliver information marked up in Wireless Markup Language (WML), so that mobile customers could participate in the Application Web. e-business flourished.

Then e-businesses wanted to integrate their processes with other e-businesses, so the *Service Web* was created. The Service Web is powered by Web application servers that speak Simple Object Access Protocol (SOAP), and deliver information marked up in Extensible Markup Language (XML). Today, the Service Web is in its adolescence. We are currently witnessing the rapid maturation and deployment of a stack of interrelated standards that are defining the infrastructure for the Service Web.

The building block of the Service Web is the *Web service*, a set of related application functions that can be programmatically invoked over the Internet. The information that an application must have in order to programmatically invoke a Web service is given by a Web Services Description Language (WSDL) document. WSDL documents can be indexed in searchable Universal Description, Discovery, and Integration (UDDI) Business Registries so that developers and applications can locate Web services.

This section describes the Service Web, outlines the key standards of SOAP, WSDL, and UDDI, and discusses new tools for developing Web services. Armed with this information, you should be able to understand how Web services can enhance your business and how you can begin developing them.

The Service Web

Web services provide an application integration technology that can be successfully used over the Internet. To illustrate this integration, consider how the Web is currently used for planning travel. Many Web sites sell airplane tickets, and most of them are well-designed for personal travel. However, they may not be useful for planning business travel, because many businesses have internal processes that control travel.

A typical business process for travel might require the use of preferred airlines, car rental agencies, hotels, and travel agents. A travel request might also require management approval to ensure that it is within corporate guidelines and expense constraints. Employees are highly motivated to follow their employer's business processes since failure to do so may result in denial of expense claims.

A business may provide a travel planning application to help employees prepare travel requests.

Suppose you are an employee using your company's travel planning application. If the request form asks you to fill in your desired flight times, you naturally want to know what flights are available. Since your travel planning application is not integrated with the airline Web site, you must launch your Web browser, go to the airline Web site, find the schedule query page, and then enter your origin, destination, and dates and times of travel. The airline Web application then returns a list of scheduled flights, from which you choose the flight you want. But since the information on the airline Web page is probably in a different format than that required by the travel application, you must write down or copy the information, switch back to the travel application, enter the information in the travel application, and then submit your request.

A simple task that should have taken seconds, instead takes minutes, because there is no easy way to integrate the airline Web application with the corporate travel planning application.

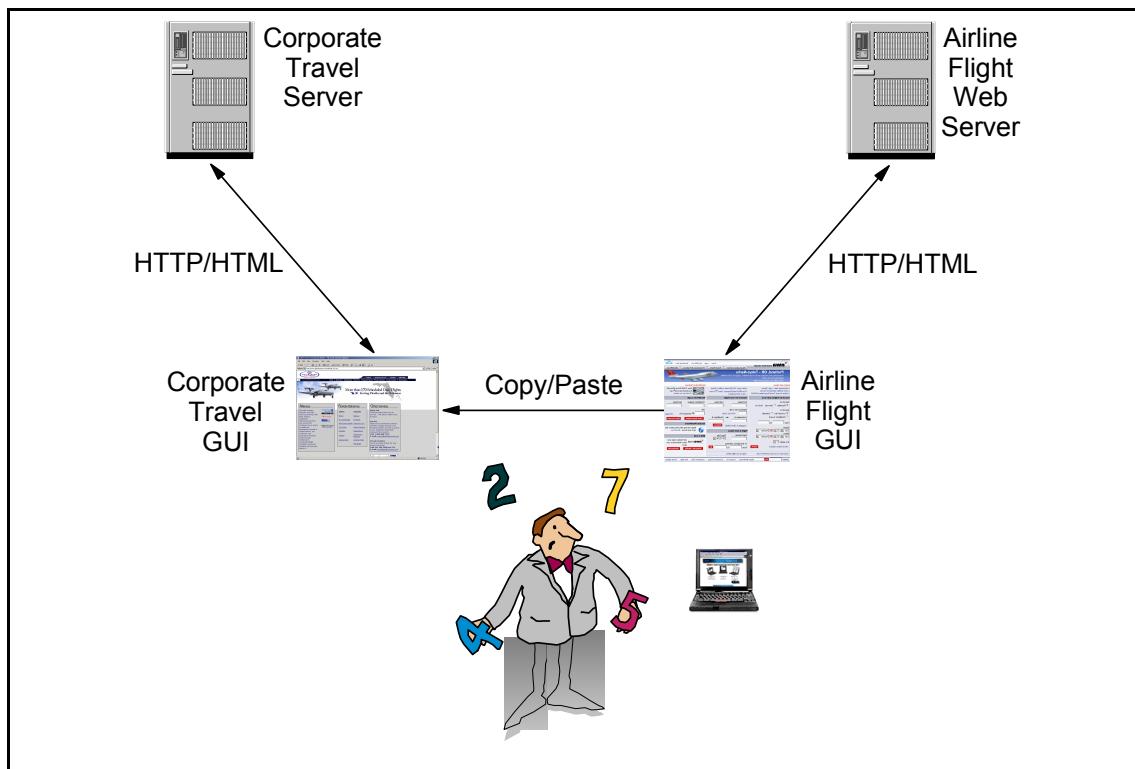


Figure A-11 Weak Integration on the Application Web

Consider how Web services could change this scenario.

Suppose the airline develops a Web service that allows applications to obtain the list of available flights between two cities on a given date. Now, the corporate travel planning application can programmatically invoke the airline flight schedule Web service, so that the employee does not have to navigate through the airline Web site. The ease of use of the corporate travel planning application is greatly improved. There is also a minor benefit to the airline since now its servers are more responsive. The Web service-based transaction is completed with a single request, whereas the traditional Web site returns many HTML pages and images to the user.

Figure A-12 shows how Web services can improve application integration.

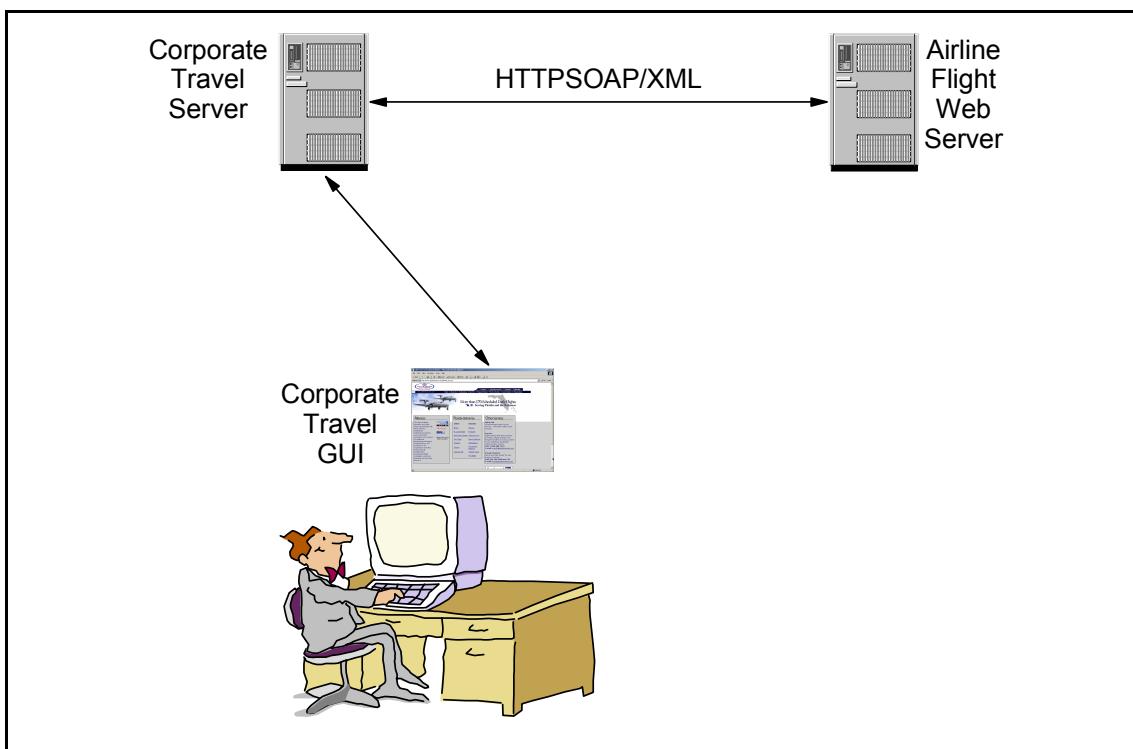


Figure A-12 Improved integration on the service Web

Continuing this scenario, suppose that airlines, hotels, and car rental agencies provide Web services that allowed applications to programmatically purchase airplane tickets, book hotel rooms, and reserve rental cars. In most cases, a travel planning application could make all of the necessary arrangements without the aid of a human travel agent.

The use of Web services would reduce costs, improve quality, and increase function. For example, suppose the airline also had a Web service that monitored flight departure and arrival times. The travel planning application could query the airline to determine if your flight was delayed, and if so, it could then notify your hotel and your car rental company to hold your reservations.

Here, the economic motivation for businesses to implement Web services is improved customer service and more efficient management of inventory. For example, if the car rental agency knows that you will be two hours late, it might be able to avoid the cost of transporting cars between agencies in order to meet the worst-case peak demand.

More interestingly, the widespread implementation of Web services would also enable a whole new wave of advanced applications. For example, suppose your flight is oversold. Your travel application calls you in the departure lounge on your WAP phone and asks what you want to do. Your travel application can access an e-marketplace Web service that lists seats on this flight. If you have a confirmed seat, your travel application asks you if you would like to put your seat up for auction, shows you the current bids, and lets you specify an asking price.

Conversely, if you don't have a confirmed seat, your travel application asks you if you want to bid on a seat, shows you the current prices, and lets you specify a bid. If you need to check your agenda in order to make a decision, your travel planning application accesses your corporate calendar application Web service and displays your agenda on the WAP phone. If you don't have a confirmed seat but really need to get to that meeting on time, the travel application verifies that corporate expense guidelines are met and approves your bid. If your bid is accepted, the travel application submits the bill to the corporate expense statement Web service.

Figure A-13 shows how powerful new applications can be assembled from the Web services of multiple suppliers.

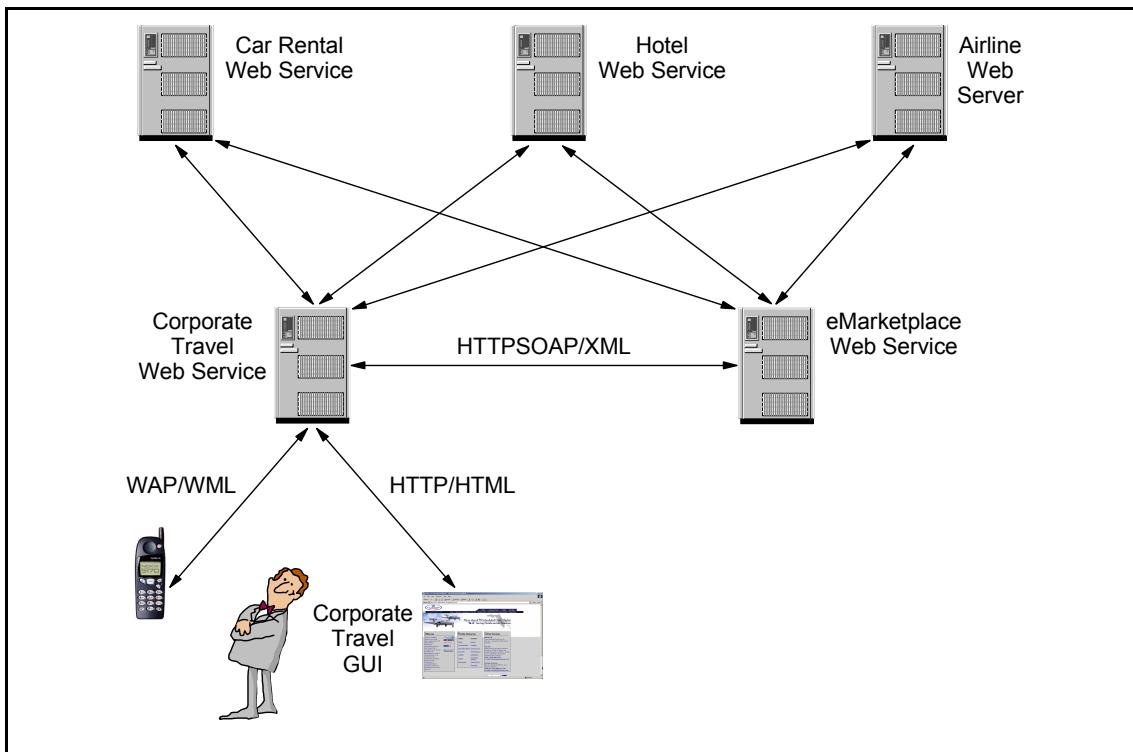


Figure A-13 The new Web services economy

As the number of Web services increases, their interactions will become more complex and we may even begin to attribute characteristics such as intelligence or self awareness to the Service Web. The usual vision of artificial intelligence as arising through some very complex thinking program must be reevaluated. More likely, intelligence may arise as a network effect. After all, the brain itself is simply a very complex network of relatively simple neurons. It is likely that the first time a collection of interacting Web services displays a useful, unanticipated behavior, we will start to think of it as intelligent.

For example, suppose you also rent a cell phone with your car, because you want to avoid long distance charges or because your phone has no local coverage. Suppose also that while driving back from your meeting to the airport, your newly rented cell phone rings and your travel application tells you that the flight has been delayed. The first time this happens to you, you might be surprised and you might think the travel application was intelligent, especially if this capability had not been explicitly programmed into it.

This type of apparently intelligent behavior could arise through the following sequence of simple Web service interactions. When you rented the car and cell phone, the travel application requested the rental agreement from the car rental agency and submitted it to the corporate expense application. When the expense application received the invoice, it scanned the items and noticed a cell phone. The expense application then updated the corporate phone directory with the cell phone number. Later, the travel application requested the departure time from the airline and noticed that it had changed. The travel application then queried the corporate phone directory, retrieved your new cell phone number, and called you.

Most travelers would consider these kinds of notifications very convenient, but other kinds of unanticipated interactions might be undesirable. As we go forth and bravely build the Service Web, we should build appropriate safeguards and controls into it in order to prevent undesired consequences. Considerations of security and privacy will be more important on the Service Web than they are today on the Application Web, and could well become the differentiators between successful e-businesses and their competitors.

Web services standards

The Service Web is being built on Internet standards. One of the key attributes of Internet standards is that they focus on protocols and not on implementations. The Internet is composed of heterogeneous technologies that successfully interoperate through shared protocols, not shared middleware. No single vendor can impose a standard on the Internet, and no single programming technology will dominate. Fortunately, Web services standards are being cooperatively developed by IBM, Microsoft, Ariba, and many others, and are being submitted to the World Wide Web Consortium (W3C).

Open source software development plays a crucial role in preserving the interoperability of vendor implementations of standards. To understand this, consider the Apache Web server, which has rapidly gained market share to become the dominant Web server. The Apache Web server acts as the reference implementation of HTTP. Consider what would happen if the next release of Microsoft Internet Explorer or Netscape Navigator included a change to HTTP that made it only work with Microsoft Internet Information Server or Netscape Enterprise Server. The market share of such a vendor-specific Web browser would drop to zero. The existence of the Apache Web server prevents any Web browser or server vendor from fragmenting HTTP. IBM is actively contributing implementations of Web services standards to open source projects in order to ensure interoperability on the Service Web.

This section discusses three major Web services standards:

- ▶ **SOAP**—A standard for messaging over HTTP and other Internet protocols

- ▶ **WSDL**—A language for precisely describing the programmatic interfaces of Web services
- ▶ **UDDI**—A business registry standard for indexing Web services, so that their descriptions can be located by development tools and applications.

Figure A-14 shows how these standards are related to each other:

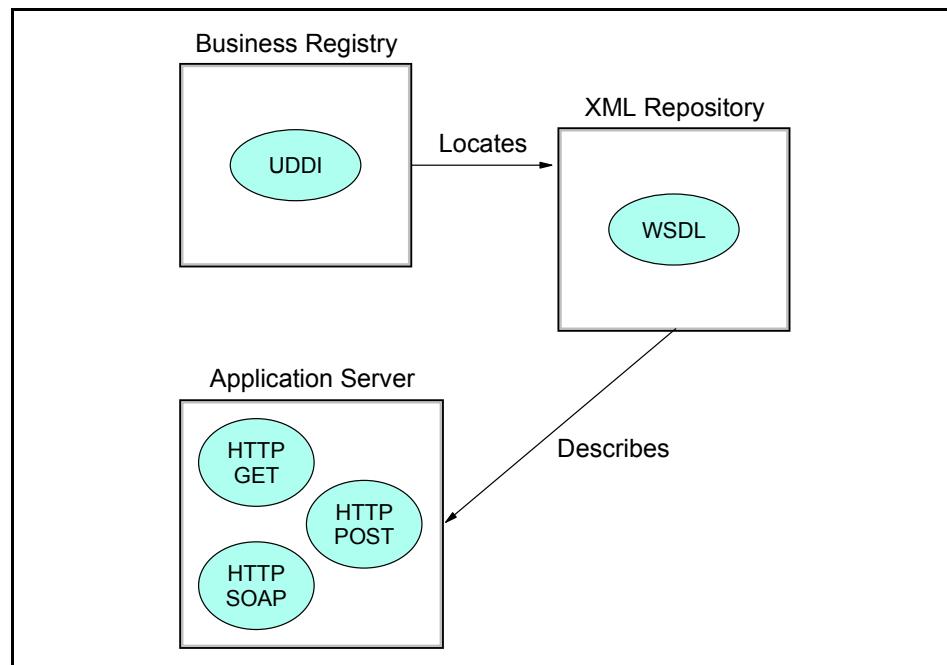


Figure A-14 How SOAP, WSDL, and UDDI are related

Application servers host Web services and make them accessible using protocols such as HTTP GET, HTTP POST, and SOAP/HTTP. The Web services are described by WSDL documents, which can be stored on the application server or in special XML repositories. A WSDL document may reference other WSDL documents and XML Schema (XSD) documents that describe data types used by Web services. An XML repository is useful for managing WSDL and XSD documents. The WSDL documents contain the URLs of the Web services. The Web services are described and indexed in a UDDI business registry that contains the URLs of the WSDL documents.

- ▶ **SOAP** is a standard for sending and receiving messages over the Internet. SOAP was initially proposed by Microsoft as a way to do Remote Procedure Call (RPC) over HTTP. IBM and other vendors contributed to the development of the SOAP 1.1 standard, which was then submitted to the W3C. The current

SOAP specification allows transports other than HTTP, and other messaging styles in addition to RPC.

The SOAP specification defines an XML envelope for transmitting messages, a method for encoding programmatic data structures as XML, and a binding for using SOAP over HTTP.

The SOAP specification defines the SOAP encoding style, which defines a way to express data structures in XML. The SOAP protocol permits the use of other encoding styles, but this is a potential source of fragmentation, because if a Web service uses a different encoding style, then its reach is limited to only those applications that support that particular encoding style. Alternative encoding styles should be used only after careful consideration of the intended users of the Web service. SOAP messages are not required to use any encoding style in which case the message body is treated as literal XML.

The SOAP specification defines a binding for using SOAP over HTTP, but other transports may be used. For example, Simple Mail Transport Protocol (SMTP) can be used to send SOAP messages to e-mail servers. Using SMTP is useful for asynchronous message delivery.

SOAP is a relatively simple protocol, and is easy to implement, as witnessed by the explosion of implementations. One of the first implementations was done in Java by developer. IBM then released another Java implementation, called IBM SOAP4J, on the IBM alphaWorks Web site. The SOAP4J implementation was later contributed to the Apache XML project and was released as Apache SOAP 2.0 which runs on Apache Tomcat, IBM WebSphere Application Server, and other servlet engines. Microsoft released a SOAP Toolkit for use with Visual Basic. Many implementations of SOAP are now available, as shown by the list of Web services at XMMethods.

SOAP has rapidly matured. In the Java world, many Java Specification Requests (JSR) were created to standardize APIs. The main JSRs are JSR 101, which defines the Java binding for SOAP, and JSR 109, which defines the deployment model for Web services. These JSRs are included in J2EE 1.4. The Apache Axis project implements JSR 101, while WebSphere Application Server V5.0.2 implements both JSR 101 and JSR 109. In the Microsoft world, Web services are supported by ASP.NET, which is part of the .NET Framework.

After the SOAP 1.1 specification was submitted to the W3C, it formed the Web services Activity to advance Web services standards, and created the XML Protocol Working Group to turn SOAP into a W3C standard. Recently, the SOAP 1.2 specification became a W3C Recommendation.

- ▶ **WSDL**—For an application to use a Web service, the programmatic interface of the Web service must be precisely described. In this sense, WSDL plays a role analogous to the Interface Definition Language (IDL) used in distributed programming. The description must include such details as the protocol, host

and port number used, the operations that can be performed, the formats of the input and output messages, and the exceptions that can be thrown.

There were several proposals for languages to solve this problem. Microsoft first proposed Service Description Language (SDL) and provided an implementation of it in their SOAP Toolkit. After reviewing the proposal, IBM countered with the Network Accessible Service Specification Language (NASSL) and released an implementation of it in SOAP4J as the NASSL Toolkit. Ideas from NASSL influenced Microsoft's SOAP Contract Language (SCL), which was also planned to describe Web service composition (also referred to as workflow or orchestration). These proposals converged with input from several other vendors as the WSDL 1.1 specification, which was then contributed to the W3C. IBM and Microsoft continued to work on Web services composition and released the Business Process Execution Language for Web services (BPEL4WS) specification.

A notable feature of WSDL is that interfaces are defined abstractly using XML Schema and then bound to concrete representations appropriate for the protocol. WSDL has predefined bindings for the protocols HTTP GET, HTTP POST, and SOAP/HTTP, but is extensible. For example, in the case of HTTP GET, the binding specifies that the input message is encoded in a URL of an XML request body.

WSDL permits the description document to be composed from parts using an import mechanism. It is useful to put the abstract definitions in a separate file, so that they can be imported by many concrete bindings. For example, the travel industry could define an abstract flight schedule Web service interface, which would then have concrete bindings by many airlines. Each airline would create its own WSDL file that specified the protocols, hosts, and port numbers used by the airline's application servers, and would import the abstract definition of the Web service from the travel industry's XML repository.

WSDL is being standardized by the W3C Web Services Description Working Group which is part of the Web services activity, and drafts of the WSDL 1.2 specification are available for public review.

- ▶ **UDDI** addresses the problem of how to find Web services. UDDI defines a business registry where Web service providers can register services, and where developers and applications can find them. IBM, Microsoft, and Ariba implemented the original UDDI registries, but other vendors now also have implementations. A service provider only has to register a Web service at one of the business registries, because updates to any registry will be replicated in all of the other registries that are part of the UDDI Business Registry Network.

Figure A-15 shows how the UDDI Business Registry Network is used by providers and users of Web services.

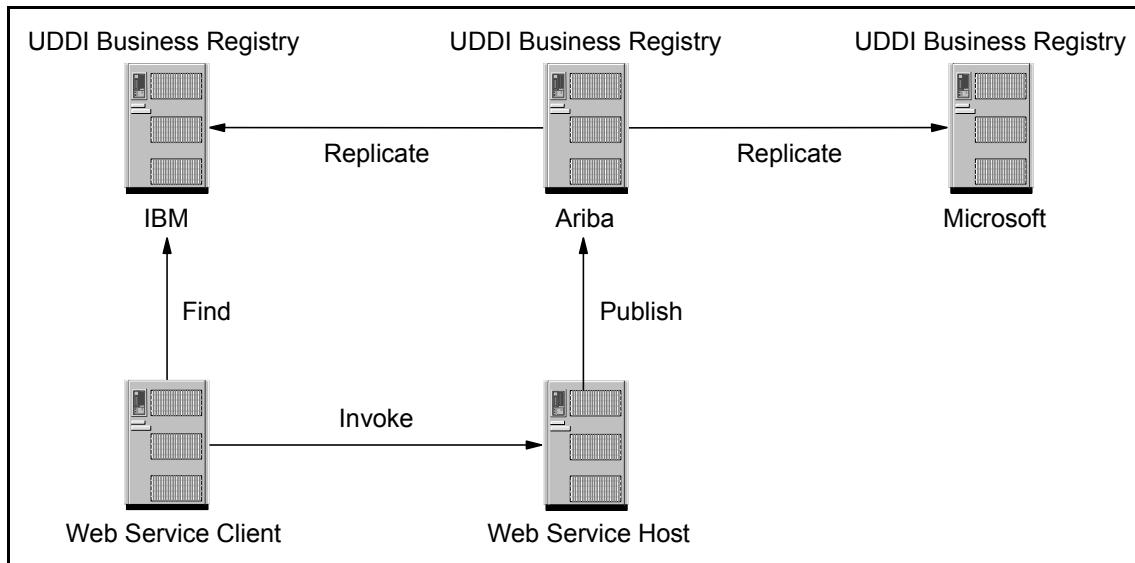


Figure A-15 The UDDI Business Registry Network

UDDI is based on four types of entity—Business Entity, Business Service, Binding Template, and Technology Model:

- A **Business Entity** describes the business that is providing the services. The Business Entity description includes categorization information so that searches can be performed for specific types of businesses.
- A **Business Service** is a class of services within a business. Each Business Service belongs to some Business Entity.
- The **Binding Template** and **Technology Model** (tModel) together define a Web service as described by WSDL. The tModel corresponds to the abstract description and the Binding Template corresponds to its concrete binding to a protocol. Each Binding Template belongs to some Business Service, but many Binding Templates can refer to the same tModel. A UDDI business registry is itself a SOAP Web service. It provides operations to create, modify, delete, and query each of the four entity types.

You can use UDDI at both development time and run time. At development time, you can search the registry for suitable services and use it to locate the appropriate WSDL file. Development tools can then be used to generate client proxies from the WSDL files so that the application can access the service. At run time, the application does not have to use UDDI if the target Web service is fixed and always available. However, if the Web service becomes unavailable, the application can query the registry to determine if the service has been moved to another location. This capability is useful when service providers have to rehost their services. UDDI can also be used at run time, for example, to determine all available services that implement some interface. For example, all car rental agencies could be located and queried at run time to find the lowest rates or the best availability at that moment.

The UDDI specification is being standardized by OASIS, along with new Web services standards for security, reliability, and other enterprise computing qualities of service.

Development tools

Although SOAP, WSDL, and UDDI form the infrastructure for Web services, developers should be shielded from their details, so that they can focus on the business problem being solved. Development tools that facilitate Web services development are now available.

Early versions of several IBM development tools, such as the WSDL Toolkit, the Web Services Toolkit, and the IBM XML and Web Services Development Environment, appeared on the IBM alphaWorks Web site, which continues to host emerging Web services technology. The IBM XML and Web Services Development Environment is now integrated into the IBM WebSphere Studio

product, which contains a full suite of Java, J2EE, Web, XML, data, and Web services tools along with tools for building, debugging, testing, deploying, and profiling applications.

Figure A-16 shows the WebSphere Studio desktop.

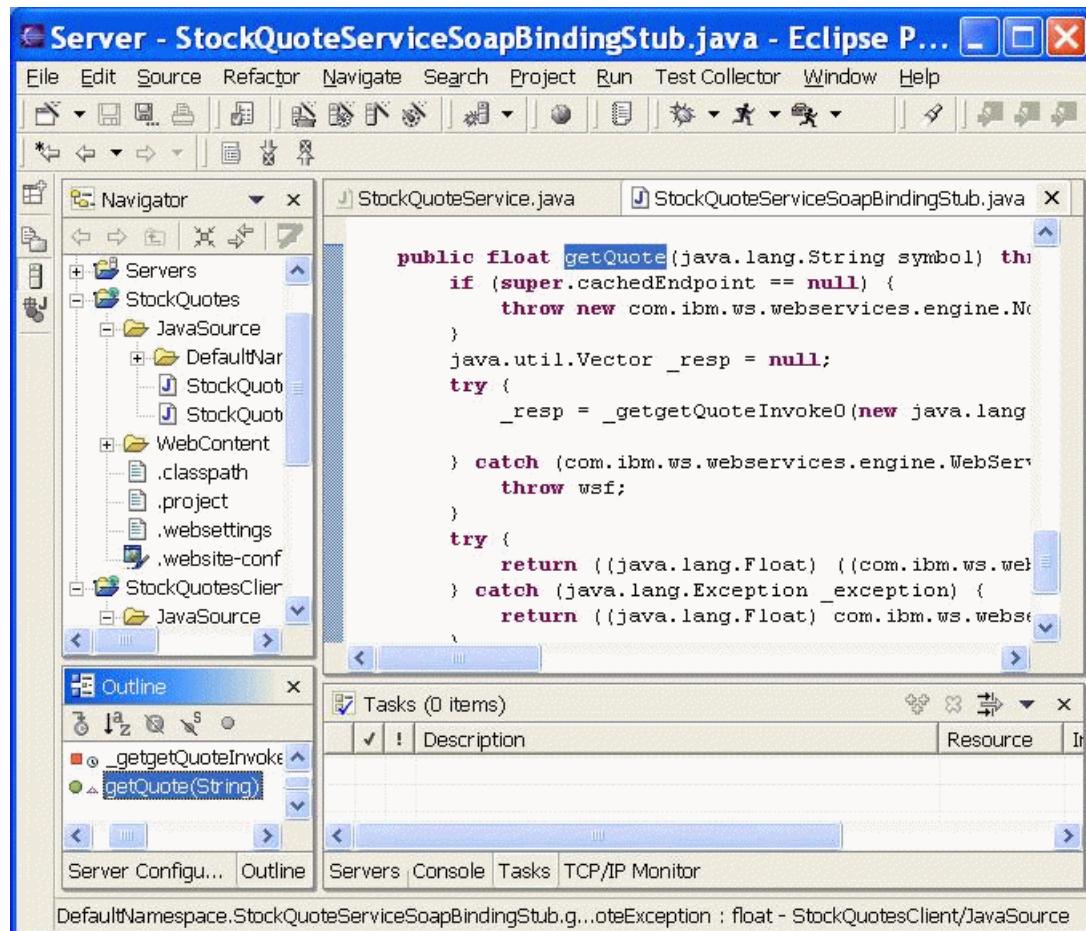


Figure A-16 WebSphere Studio desktop

The Navigator pane shows the projects, folders, and files in the currently open workspace. The Editor pane contains two open files: StockQuoteService.java and StockQuoteServiceSoapBindingStub.java. Many files may be open at once, but only one at a time can be selected for editing. You select the files using the tabs at the top of the pane. When a file is selected, its contents are displayed and it is outlined in the Outline pane. Parts of the file can be selected by clicking in the Outline pane. The All Tasks pane lists problems in the solution.

Double-clicking on a task opens the file and scrolls it to the line where the problem occurred.

WebSphere Studio has a built-in Java development environment and a set of XML and Web services development tools. The XML tools include editors for XSD and WSDL. The Web services tools include a wizard that can create a Web service from a Java class, an EJB, or a set of SQL statements. The wizard can then generate a WSDL file that describes the Web service, generate a Java client proxy and server skeleton from a WSDL file, generate a Web test client from a WSDL file, and explore the Web services in a UDDI Business Registry. The Web services wizard frees you from dealing with the details of XML, SOAP, WSDL, and UDDI, and lets you focus on the business problem at hand. For more information on the Web services tools in WebSphere Studio, see the *IBM Systems Journal* article or the book chapter in *Java Web Services Unleashed*.

Interoperability

The key value business value of Web services is that they enable interoperability of applications that use different hardware, operating systems, programming languages, or middleware. While SOAP, WSDL, and UDDI are a good starting point, they contain too many features to ensure consistent implementation and interoperability. For example, early experience with SOAP exposed problems in achieving interoperability when using SOAP encoding. To ensure that Web services were interoperable, IBM, Microsoft, and other vendors formed the Web Services Interoperability Organization (WS-I.org). The membership of WS-I.org includes all major vendors and many customers, so there is good reason to believe that interoperability will become a reality in the near future.

WS-I.org established a Basic Profile 1.0 that clarifies the specifications for SOAP, WSDL, and UDDI, and identifies a subset of features that can be implemented by all vendors in an interoperable way. The Basic Profile 1.0, which covers SOAP, WSDL, and UDDI, is in the final stages of approval and is part of J2EE 1.4. WebSphere Application Server V5.0.2 and WebSphere Studio V5.1 will enable the creation of WS-I.org compliant Web services. The Basic Profile 1.1 will include support for SOAP with Attachments and WS Security. SOAP with Attachments allows binary documents, such as images, to be efficiently transported with XML SOAP messages. WS Security defines how portions of SOAP messages can be digitally signed or encrypted. As new key Web services standards emerge, they will be added to future WS-I.org profiles.

The next generation of advanced Web applications will be constructed from Web services building blocks. By developing Web services, businesses enable their applications to be integrated into the processes of their suppliers, partners, and customers, yielding improved function, reduced costs, and competitive advantage.

Web services are based on open, implementation-neutral standards such as XML, SOAP, WSDL, and UDDI. The Web services infrastructure is now ready for serious consideration by businesses.

Development tools that support Web services standards are available and IBM has a full-featured suite of Web services tools in WebSphere Studio.

Build interoperable Web services with JSR-109

Note: The following material comes from “Understanding the foundation of JSR-109” by Jeffrey Liu and Yen Lu, (IBM software developers), published August 2003. IBMers can find this at <http://www-106.ibm.com/developerworks/webservices/library/ws-jsrart/>.

JSR-109 facilitates the building of interoperable Web services in the Java 2 Platform, Enterprise Edition (J2EE) environment. It standardizes the deployment of Web services in a J2EE container. This article discusses the server and client programming models defined by JSR-109 and provides code examples.

A crucial objective of Web services is to achieve interoperability across heterogeneous platforms and runtimes. Integrating Web services into the Java 2 Platform, Enterprise Edition (J2EE) environment is a big step forward in achieving this goal. JAX-RPC (JSR-101) took the first step into this direction by defining a standard set of Java APIs and a programming model for developing and deploying Web services on the Java platform. JSR-109 builds upon JAX-RPC. It defines a standard mechanism for deploying a Web service in the J2EE environment, more specifically, in the area of Enterprise JavaBean (EJB) technology and servlet containers. Both of these specifications will be integrated into the J2EE 1.4 specification. Together they serve as the basis of Web services for J2EE.

- ▶ **Roles in Web services for J2EE development**—The J2EE platform defines several roles in the application development cycle. They include: J2EE product provider, application component provider (developer), application assembler, deployer, system administrator, and tool provider. In an attempt to integrate Web services development into the J2EE platform, JSR-109 defines additional responsibilities for some of the existing J2EE platform roles. The J2EE product provider is assigned the task of providing the Web services runtime support defined by JAX-RPC, Web services container, Web services for J2EE platform APIs, features defined by JAX-RPC and JSR-109, and tools for Web services for J2EE development. In the actual Web services for J2EE development flow, the developer, assembler, and deployer are assigned specific responsibilities.

Figure A-17 illustrates the Web services for J2EE development flow.

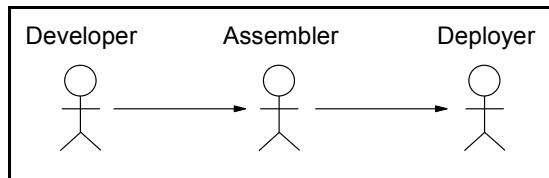


Figure A-17 Web services for J2EE development flow

In general, a developer is responsible for providing the following:

- Web service definition
- Implementation of the Web service
- Structural information for the Web service
- Implementation of handlers
- Java programming language and WSDL mappings
- Packaging of all Web service related artifacts into a J2EE module

Figure A-18 on page 298 summarizes a developer's responsibilities in the server and client programming models. We will present further details on each of these programming models later in this article. Figure A-18 on page 298 also illustrates which specific resources each of these responsibilities touches upon. For example, the Web service definition is provided in the form of a WSDL file, the structural information of the Web service and the Java-to-WSDL mapping are defined in webservices.xml and jaxrpcmapping.xml, respectively. These two XML files are deployment descriptors defined in the server programming model.

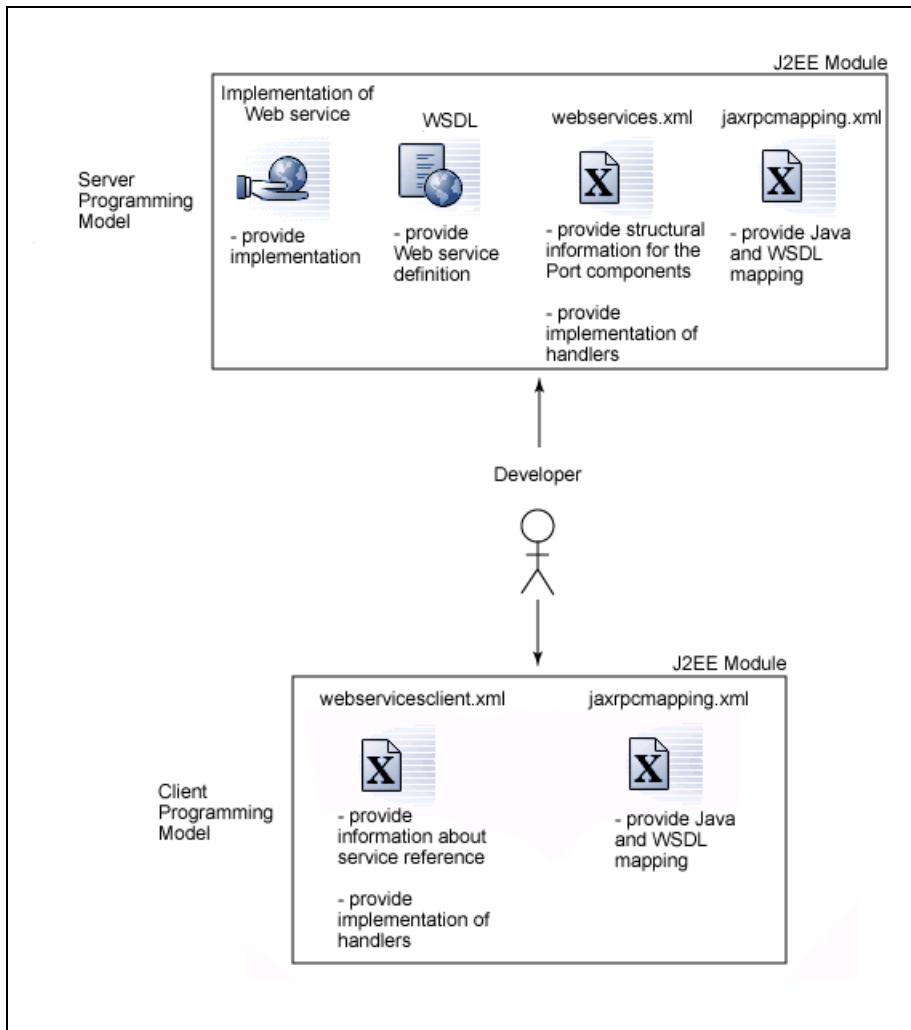


Figure A-18 Web services developer role

An assembler takes the J2EE modules developed by a developer, and assembles them into a complete J2EE application. An assembler is responsible for the following:

- Assembling modules into an Enterprise Application Archive (EAR)
- Configuring the modules within the application, for example, resolving module dependencies
- Configuring handlers

Figure A-19 on page 299 summarizes an assembler's responsibilities.

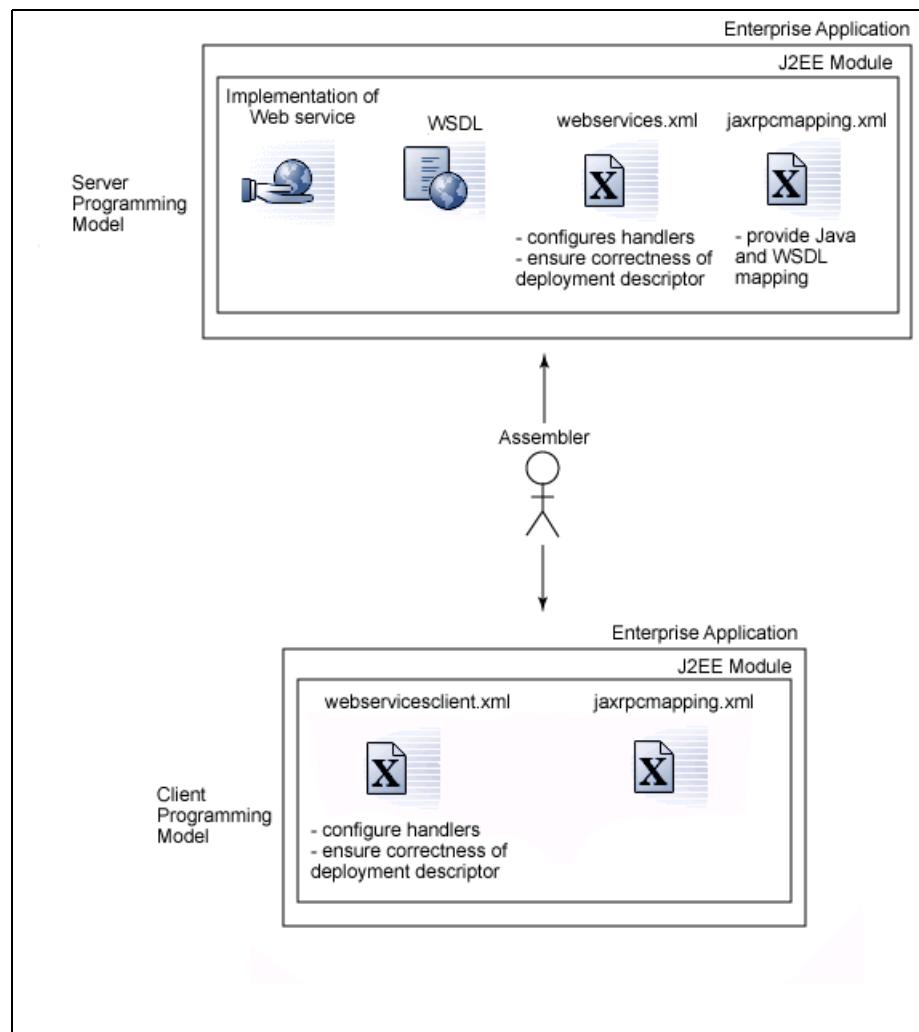


Figure A-19 Web services assembler role

A deployer is responsible for the following:

- Resolving endpoint addresses for Web services contained in the EAR
- Generating stubs for Web services using deployment tools provided by a Web Services for J2EE product provider
- Resolving the WSDL documents used for each service reference
- Resolving port access declared by a port component reference
- Deploying the enterprise application

Figure A-20 on page 301 summarizes a deployer's responsibilities.

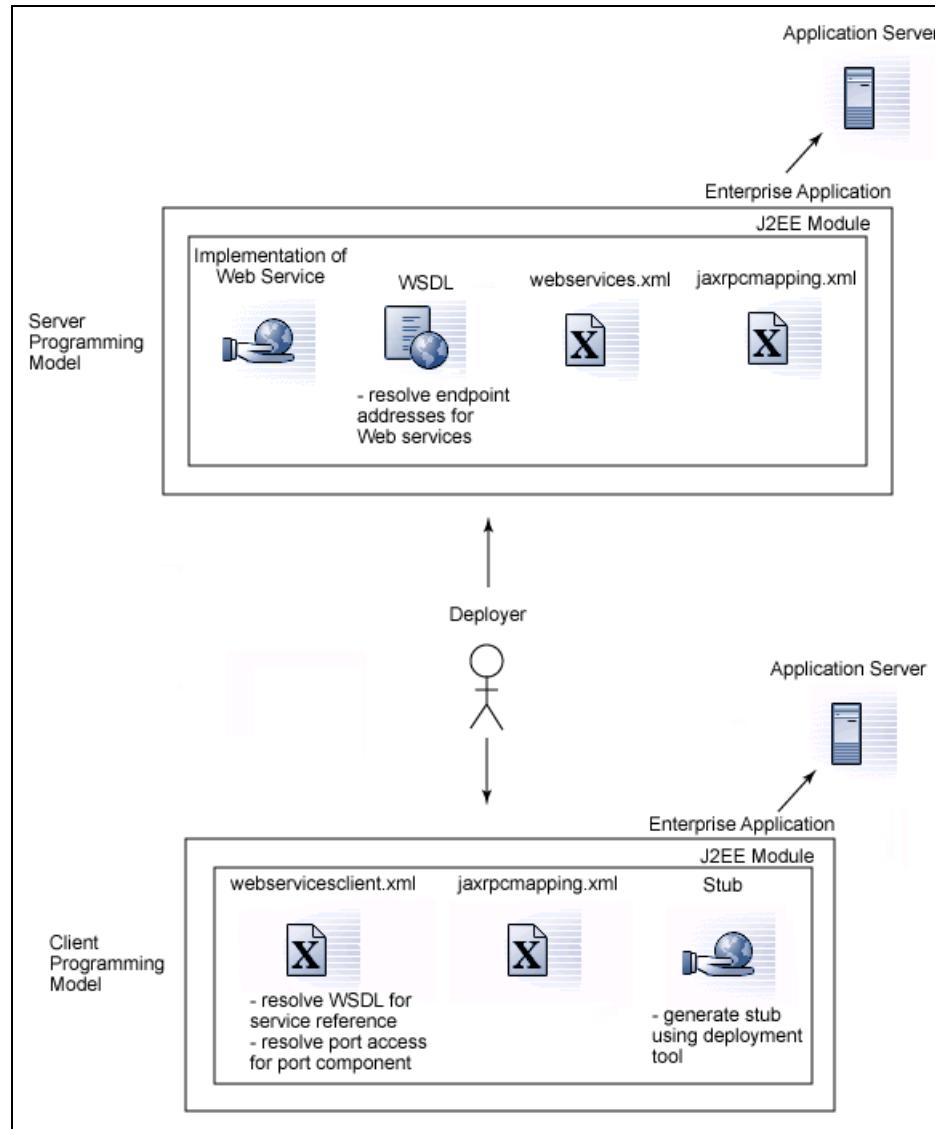


Figure A-20 Web services deployer role

- ▶ **Benefits of JSR-109**—Prior to JSR-109, the procedure for deploying a Web service was highly coupled to the destination runtime. Deploying a Web service to Apache Axis is quite different from deploying to Apache SOAP. Organizations that have deployed Web services in the past would require some strong arguments as to why they should switch to the JSR-109 deployment model given their previous investments. The motivation for

JSR-109 is to promote the building of interoperable and portable Web services in the J2EE environment. JSR-109 leverages existing J2EE technology and provides industrial standards for Web services deployment. It clearly defines requirements that a Web service for J2EE product provider must support. It allows a J2EE Web service to be configurable via XML-based deployment descriptors and hides the complexity of the system from Web service developers, assemblers, and deployers. Knowing how JSR-109 works allows you to configure a J2EE Web service without having to explore and learn the implementation details of the underlying system. Finally, as JSR-109 is adopted by Web server providers, the process of migration and deployment should become a routine procedure.

- ▶ **Server programming model**—The server programming model attempts to standardize the deployment of Web services in the J2EE environment. More specifically, it defines the deployment of a servlet-based implementation bean in the Web container and the deployment of a stateless session EJB implementation in the EJB container. Both types of service implementations use a Service Endpoint Interface (SEI)—refer to JAX-RPC (JSR-101)—to define the method signatures of the Web service. The SEI must follow the Java programming language and WSDL mapping rules defined by JAX-RPC. The service implementation bean must implement all the methods defined in the SEI. For EJB service implementation, the SEI methods must be a subset of the EJB component's remote interface methods. The lifecycle of the service implementation bean is controlled by its associated container. In general, a service implementation bean will go through four phases: the instantiation phase, the initialization phase, the execution phase, and the removal phase. The lifecycle starts off when the container creates an instance of the service implementation bean by calling its *newInstance* method. The container then initializes the bean via its container specific method. For a servlet-based implementation bean, the *init* method is called. For an EJB implementation, the *setSessionContext* and the *ejbCreate* methods are called. The service implementation bean is now ready to handle service requests. The lifecycle ends when the bean is removed from the container. This will result in the *destroy* and *ejbRemove* methods being called for the Web and EJB implementations, respectively. Figure A-21 on page 303 shows the lifecycle of a servlet-based implementation bean and Figure A-22 on page 303 shows the lifecycle of a session EJB component.

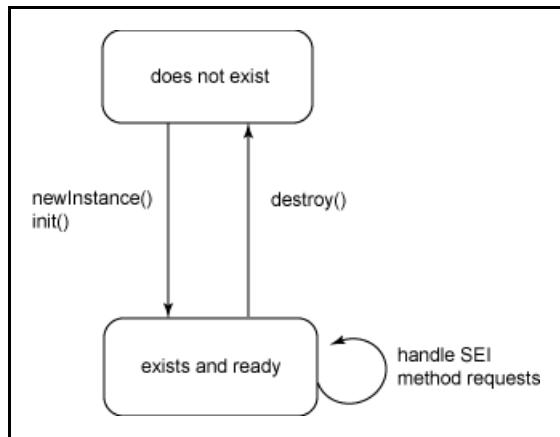


Figure A-21 Lifecycle of a servlet-based implementation bean

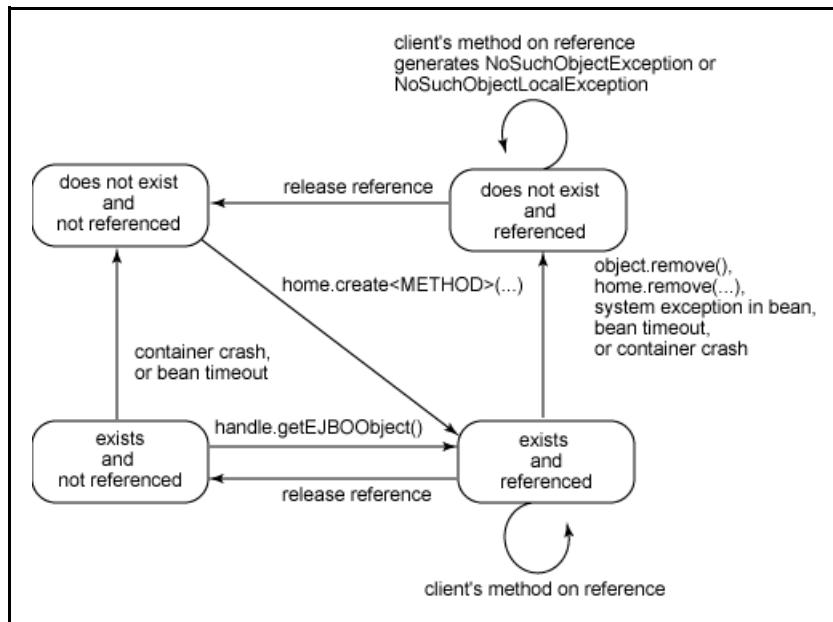


Figure A-22 Lifecycle of a session EJB

IBMer can find more information at
<http://www-106.ibm.com/developerworks/webservices/library/ws-jsart/>.

- ▶ **Client programming model**—The client programming model provides a client view of a Web service in the J2EE environment. Run time details, such as protocol binding and transport are transparent to a client. A client invokes

a Web service in the same way as invoking a method locally. JSR-109 specifies three mechanisms for invoking a Web service:

- Static stub
- Dynamic proxy
- Dynamic Invocation Interface (DII)

Static stub is a Java class that is statically bound to an SEI and to a WSDL port and port component. It is also tied to a specific protocol binding and transport. A stub class defines all the methods that an SEI defines. Therefore, a client can invoke methods of a Web service directly via the stub class. It is extremely easy to use and requires no configuration. However, the major disadvantage of a stub class is that the slightest change of a Web service definition may be enough to render a stub class useless. In this case, the stub class must be regenerated. A stub class is very handy in situations where you want to create a client to a production Web service that is very stable and is not likely to change its definition.

Dynamic proxy, as its name suggests, supports an SEI dynamically at run time without requiring any code generation of a stub class that implements a specific SEI. A dynamic proxy can be obtained during run time by providing an SEI. It is not as statically bound as a stub class, but it requires an SEI to be instantiated. It is invoked in the same way as a stub class is invoked. A dynamic proxy can be very useful during the Web services development cycle. A Web service's definition—more specifically, its SEI—may undergo a lot of changes over a period of time. Using a stub class to test the Web service could be very tedious, because a stub class needs to be regenerated every time the SEI changes. In this case, a dynamic proxy is more suitable because it only needs to be reinstantiated (rather than regenerated as in the case of a static stub) whenever the SEI changes.

IBMer can find more information at

<http://www-106.ibm.com/developerworks/webservices/library/ws-jsrart/>.

- ▶ **JAX-RPC mapping**—The last piece to the JSR-109 puzzle is the mechanism for standardizing the Java <=> WSDL mappings. JAX-RPC provides the rules for these mappings, while JSR-109 provides a XML-based deployment descriptor standardized representation.

IBMer can find more information at

<http://www-106.ibm.com/developerworks/webservices/library/ws-jsrart/>.

- ▶ **Future improvements**—JSR-109 provides a preliminary standardized mechanism for deploying Web services in the J2EE environment. There is still room for future improvements, especially in the area of security. Although the demand for secure Web services is as important as having interoperable Web services, the challenge to standardize a security model across heterogeneous platforms remains. As for JSR-109, it defines the security requirements that it attempts to address, but the actual standardization is

deferred to a future release of the specification. Security requirements include the following:

- Credential-based authentication. For example, HTTP BASIC-AUTH.
- Authorization defined by the enterprise security model.
- Integrity and confidentiality using XML encryption and XML.
- Digital signature.
- Audit.
- Non-repudiation.

Besides security improvements, there are areas where JSR-109 can improve that are directly related to JAX-RPC. For example, JAX-RPC defines an in-depth Java <=> XML serialization and deserialization framework. However, JSR-109 does not provide any support in this area. In addition, JSR-109 did not provide a complete representation for the type mapping rules defined by JAX-RPC. It lacks the support for MIME types. For example, JAX-RPC allows `java.lang.String` to be mapped to MIME type `text/plain`. However, JSR-109 cannot support this mapping, as there is no standard representation for MIME types in the JAX-RPC mapping file.

Hopefully, some of these issues will be addressed in a future release of the JSR-109 specification. This will provide a more integrated, interoperable, and portable architecture for building Web services.

User interface integration

Note: The following is from “The Case for Portlets,” by Ann Marie Fred and Stan Lindesmith (IBM staff software engineers), published February 1, 2003. IBMers can find this article

<http://www-106.ibm.com/developerworks/ibm/library/i-portletintro/>.

Java servlets, Java Server Pages (JSP), and Microsoft Active Server Pages (ASP) technologies are the tried-and-true development platforms for Web applications. They are mature platforms, with accepted standards and plenty of tools for developing and testing them.

Servlets are essentially a standard way of extending the functionality of a Web browser. They were designed to replace CGI (Common Gateway Interface) scripts. There is a standard API (application programming interface) that developers “fill in” to implement their Web applications. The servlet engine provides some useful and time-saving features, including converting HTTP requests from the network to an easy-to-use `HttpServletRequest` object, providing an output stream for the programmer to use for the response, and converting the easy-to-use `HttpServletResponse` object to an HTTP response that can be sent back over the network. The servlet engine also provides convenient session management features that allow you to remember users

between requests, and you can allocate resources (such as database connections) that can be used for multiple requests. Like portlets, servlets act as a bridge between your business data and a Web browser interface. Servlets have the advantage of maximum portability: they can run within more Web servers or Web application servers and on more platforms than any other Web application technology available today.

JSP pages are actually a type of servlet. Each JSP page is automatically converted into a servlet behind the scenes. Just like a servlet, JSP pages give developers access to the request and response objects, and you write your output to the response's output stream. Some developers make the mistake of completely replacing their servlets with JSP pages, but ideally they are used differently and complement each other. JSP pages provide a convenient syntax for generating your output—a mixture of Java, special JSP syntax, and browser markup (like HTML and JavaScript)—while servlets, which are pure Java technology, provide a cleaner way to interact with your business data. The Model-View-Controller design, which is familiar to many developers, helps us to use servlets and JSP pages together. The Model is your business data and objects that help you read and update it. The Controller is the servlet, which takes requests from the user's Web browser, acts on them using the Model, and neatly packages the information that the JSP page will need into one object (usually called a bean). The JSP page is the View, which renders the information from the servlet in a clear and visually pleasing way. Any platform that supports servlets can also support JSP pages.

ASP pages provide functionality similar to servlets and JSP pages. ASP pages are used to create an interactive Web page experience for the user. ASP pages also enable developers to use an MVC design to separate the business logic from the presentation by using ActiveX controls. They also integrate easily with Microsoft's .NET platform. However, unlike servlets and JSP pages, ASP pages are not compatible with Java portlets, and there is not an easy way to convert them into portlets. Because of the ActiveX controls, ASP pages are restricted to Windows platforms (unless you can find a third-party porting/bridging program that exists for the platform you are targeting). Overall, ASP pages are not as compatible with nearly as many Web servers, Web application servers, and operating systems as the Java technologies, so we prefer servlets and JSP pages.

Understanding portlets

Portlets are a hot new Web application development technology. Although JSR-168 provides a portlet specification, portlets are not fully standardized yet. For example, WebSphere Portal Server portlets will not work in the BEA portal server because the BEA application server does not support the WebSphere Portlet API. There are also not as many tools available to help you develop portlets as there are for the more established technologies like servlets.

However, this is not a reason to avoid them. Once the standards are finalized, it should be simple to convert existing portlets to meet the new standards. Toolkits for developing them are already available, and development tools are improving constantly. Check out the developerWorks WebSphere Studio Zone for portal and mobile device toolkits for WebSphere Studio.

There are not nearly as many trained portlet developers as there are experienced servlet, JSP, and ASP developers. However, developers who have experience with these older technologies, especially servlets and JSP technologies, can quickly and easily learn how to write portlets. It is more cost-effective (and less tedious) for developers to write portlets than servlets because the portal server provides many features that the developers would otherwise have to provide themselves.

Portlets are actually a special type of servlet, and they also use JSP pages to render their user interfaces. The Portlet API extends and subclasses the Servlet API, meaning that portlets can do anything that servlets do, with some changes in their behavior and some extra features. The most significant behavior change is in how requests are served: servlets process *doGet* and *doPost* requests, which map to HTTP GET and POST requests, while portlets process *doView* and *doEdit* requests, which come from the portal server instead of directly from a Web browser. A minor improvement is the *PortletLog*, which provides better logging and tracing features than the standard Servlet API. A second improvement provided by the WebSphere Portal Server is better support for internationalization of your portlets. The Portlet API also gives you access to configuration information about your portlet application and your portlet instance.

Portlets also have many standard features that are not available to servlets at all. One key feature is built-in support to automatically use different JSP pages for different user devices. This lets you write portlets that work on many devices, including desktop computers with modern Web browsers, older or palmtop computers with limited Web browsers, personal digital assistants (PDAs), and Web-enabled wireless phones. You do not need to program to a lowest common denominator; you can provide multiple interfaces for different device types, reusing the same underlying business logic, and the portal server will choose the most appropriate one for each user. You can even have multiple portlet controllers, which allows you to have different page/action sequences for each device type.

Another important feature provided by the Portlet API is the ability to handle messages and events. The *ActionListener* turns user clicks in your pages into action events, an easy-to-use model familiar to many Java programmers. The *WindowListener* notifies your portlet of window events from the portal, such as when your portlet is maximized, minimized, and restored. The *MessageListener* lets you handle messages sent from one portlet to another.

Yet another key feature of portlets is the ability to customize them in a number of ways. When developers use the standard Cascading Style Sheet provided by the portal server, all of their portlets will have the same look and feel. Companies can create themes and skins for their portlets that reflect their individual style, and all of the portlets (even those developed by a third party) will change to that new style automatically.

You can also customize your portlets on a user group basis. You can assign some user groups permissions to use portlets, and those without permissions to use the portlets will not even see them in their portal. Your portlet can also change its output based on the user group(s) that the current user is in because the portal server provides an easy way to determine group membership. The portal server can use your existing LDAP directory to import user groups, or you can create a new user directory.

The third type of portlet customization is user preferences. Subject to limits that the portal administrator sets, users can decide which portlets are in their portal, what pages those portlets are on, and any other preferences your developers allow them to change. For example, you can allow your users to customize their stock ticker portlet with their own portfolio of stocks, and store that information in the *PortletData* object for that user. When the user logs out and returns they will have their saved preferences from the previous session.

There are many other valuable features provided by the WebSphere Portal Server and its extensions, including content management, transcoding, voice support, and offline browsing. If you choose to develop portlets you will also have these features available to you, but they are outside the scope of this article. More information on these advanced features is available on the WebSphere Portal Server home page and the developerWorks WebSphere Portal Zone.

Portlets and Web services

Web services are another flashy new Web application development technology. Web services are modular applications that can be described, published, located, and invoked over the Web. These modular applications are loosely coupled, dynamically bound components. For a good introduction to Web services and how they can be applied to real world business problems, see the series "Best practices for Web services" (developerWorks, January 2003).

Like portlets, Web services are not fully standardized yet, but standards work is well underway, and you can use Web services today. There are also some development tools available already, including the WebSphere SDK for Web Services.

A Web service is a logical extension of object oriented design. A developer can implement a function and then define that function (via an API) as a Web service.

After the application is created and the interfaces defined, the Web service is made available (published) so that a customer (requester) of the service can use it. The architecture has three distinct users—a *Web service provider*, who defines the modular application and then asks the *Web service broker* to publish the application so that a *Web service requester* can use it in their application.

When do portlets make the most sense?

If your goal is to bring together your Web applications and information into one convenient place, portlets are the obvious choice. If your development goals are somewhat different, consider these other portlet and portal server features that you might want to take advantage of:

- ▶ Portlets can be extended to work on many client devices. Your users can move from computer to computer, and mobile device to mobile device, and still use the information and applications they need.
- ▶ Portlets allow you to easily customize their content for different user groups, and individual users can rearrange and tailor them to their needs.
- ▶ You can make your portlets have a unified look, and change their appearance quickly, using Cascading Style Sheets along with themes and skins that the portal server provides. You can create your own themes and skins as well, to better reflect your company's image and style.
- ▶ Portlets can be published as Web services, so that companies outside of your portal server's environment can easily write programs to use them.
- ▶ The WebSphere Portal Server provides excellent support for internationalization, beyond what your Web application server provides. It is straightforward to develop portlets that will display correctly for international users, even in double-byte or bi-directional languages like Chinese and Arabic.
- ▶ Portlets help divide complex applications into tasks: in general, one group of closely related tasks equals one portlet. WebSphere Portal Server's administration portlets are a good example, where administration tasks are broken down into categories (Portlets, Portal Settings, etc.), groups of related tasks (Manage Users, Manage User Groups), and single tasks (Search for users, Create new user).
- ▶ Portlets make it easy to add features to your applications later. If the new feature is large, you should create a new portlet. For small updates, you can update the existing portlets without losing users' individual preferences.
- ▶ Portlets, like other Web applications, play well with firewalls. They use standard Web protocols to receive and display information.
- ▶ You only need to install and configure portlets once for all of your users, which is much easier than working with standalone applications on each computer. This applies to the other Web application technologies as well.

- ▶ The portal server works with the Web application server to provide security, installation support, reliability, and availability for many users, so you don't need to spend a large part of your development effort working on these features.
- ▶ Once you do invest in a portal server, you may find its advanced features useful: content management, transcoding, voice support, and offline browsing, among others.

When are other solutions better?

Portlets are not the solution to every design challenge. Here are a few things that portlets do not do well:

- ▶ Complex user interfaces do not translate well to portlets. The markup languages like HTML and WML simply cannot describe some interfaces. Try to imagine implementing an Integrated Development Environment (IDE) like Eclipse or Visual Basic in HTML and you'll have the idea. Native applications and Java applications work better for this. (If you have a complex user interface and still want to take advantage of the benefits of portlets, WebSphere Portal Version 4.2 does support Struts, which can be very helpful. Struts are part of the Apache Jakarta project—more information is available on the Struts site.)
- ▶ User interfaces with data that must be constantly updated are also not portlet material. When you update one portlet, all portlets on the entire page must be re-drawn, so it is generally not a good practice to have your portlets automatically reload themselves with new data. On the other hand, you can have the *refresh* option in the portlet so your users can choose when to reload the page.
- ▶ Highly interactive user interfaces do not translate well to Web applications in general, or portlets in particular. If you want your interface to change automatically when a user takes some action, like selecting an entry in a drop-down list, you can either submit the form and reload the entire page (annoying), or use a scripting language to re-draw the portlet (very difficult). If you use a scripting language, you will need to make sure it works for all of the devices you want to support, and you will also need to make sure your portlet still works if scripts are disabled by some of your users. For mobile devices, you will probably need to have alternate JSP pages that do not use scripts. Native applications or Java applications are easier to make highly interactive than are Web applications.
- ▶ Portlets need to live “within their box.” Be careful if you have a link in a portlet that takes you to a Web page outside of the portal server environment, because it is difficult to get back to the portal after that. Frames are not allowed (Internal frames are allowed, but only Microsoft Internet Explorer users can see them). Pop-up windows and scripts usually cannot be used for

mobile devices. If you can't make your application fit into the portal framework, don't make it into a poorly behaved portlet.

- ▶ If you will want to provide services to other applications, consider writing a Web service first. Once you implement a Web service, you can write a portlet to use it, and you can publish the Web service to share it with other applications. The stock portlet above is a good example: the stock quote service should be a Web service that the stock quote portlet and other applications can use. In this case, you might also write a program that automatically sends users a text pager message when a stock reaches a certain price.
- ▶ If your company does not have a portal server yet, and does not plan to invest in one immediately, go ahead and implement your application as a servlet using JSP pages for the output. You can always convert it to a portlet later.
- ▶ Portlets are not fully standardized, and they are not yet supported on as many platforms as the other Java technologies. Until then, you will have to lock yourself into one portal server on the server platforms that it supports. If this is not an acceptable trade-off, implement your application as a servlet first and you will be able to convert it to a portlet later.

Knowledge management and integration

Note: The following is from “A preview of Lotus Discovery Server 2.0” by Wendi Pohs and Dick McCarrick, published in May 2002. IBMers can find this article at <http://www-10.lotus.com/ldd/today.nsf/lookup/DSpreview>.

Few would dispute that knowledge ranks among an organization's most precious assets—if you can find it when you need it. But knowledge is not always easy to locate. Ideally, combined intellect and experience of the entire group should be readily available to everyone all the time.

With some of the tools available today, you can search for all the information your organization or the Internet has collected on a topic, but too often the results are disappointing. The information is too voluminous, unfocused, or out of context—or simply too difficult for the average person to understand. What users really need is experience—the perspective of someone who has done it before. That experience could come in the form of a document, or (even better) a person to whom you can look for assistance. So what's needed is a product that speeds the knowledge management process in a way that brings new scalability to the task of analyzing your ever-growing volume of content. It should also eliminate

the need to manually process and manage knowledge. The Lotus Discovery Server answers these needs.

- ▶ The **Lotus Discovery Server** is a back-end server for managing your organization's knowledge. The Discovery Server provides sophisticated tools that categorize documents and user information into browsable and searchable form. These tools include the following.
- ▶ The **Knowledge Map** or *K-map* (also called the *taxonomy* or *catalog*) is a graphical representation of your organization's knowledge. It displays a hierarchical set of categories and documents you can use to find information. The K-map is the backbone of the Discovery Server search-and-browse user interface. From the K-map interface, you can locate content from many disparate sources, by drilling down through subject categories, using full-text search, or using a combination of both search strategies. Additional information about the relationships between people and document activity adds value and context to the user's search and retrieval experience. Because the K-map displays related documents, people, and places in categories, users can browse and search for information in context.
- ▶ The **K-map Building Service** creates the K-map, which you can subsequently modify using the K-map Editor (explained next). The K-map Building Service builds document categories, creates labels for these categories, and places new documents into existing categories. It also identifies documents that do not fit into any existing categories.
- ▶ The **K-map Editor** is a client application that lets you fine-tune the K-map to meet the needs of your organization. Neither the K-map Building Service nor any other automatic process can predict precisely how an organization wants to structure its content. It can only build a K-map based on the words in the content. Once the basic K-map is built, you can use the K-map Editor to drag categories from one level to the next, re-label them with preferred terms, and place documents in different categories. This in turn helps teach Discovery Server how to categorize documents with similar content in the future.
- ▶ **Profiles** help identify the right people for the right job. Profiles collect existing user information from the directory and other sources, providing a more complete representation of the users in your organization.
- ▶ **Spiders** are multithreaded processes that collect data. This data can exist in a number of different file formats, including:
 - XML
 - Exchange e-mail and public folders
 - Web content
 - Windows-compatible operating system files
 - Lotus Team Workplace
 - Domino.Doc®
 - Lotus Notes databases and e-mail

Once the spiders collect this data, the K-map Building Service processes it to create the K-map.

- ▶ **Metrics**, which is also called affinities processing, is a computational program that looks at existing documents and relationships between documents and people. The metrics component does two things. First, it calculates the value of a document. Second, it calculates an affinity between a person and categories, based on the person's interactions with documents in the categories, which in turn helps produce category affinities.
- ▶ **Other tools**—Additionally, administration tools let you install, set up, and maintain Discovery Server, and security features protect your data.
- ▶ **What's new in Discovery Server 2.0?**—A revamped architecture for the K-map user interface and editor, with more accessibility and better display of search results
 - Improved “people awareness” for Profiles
 - Better spider support for Domino.Doc and Lotus Team Workplace content
 - Enhanced control over metrics processing
 - Easier installation, setup, and maintenance
 - Better logging
- ▶ **K-map: Improved interface, expanded features**—The 2.0 K-map offers a user interface that features a new servlet-based architecture. This gives you more scalability and better performance. Other K-map enhancements include accessibility, improved display of search results, bookmarking, and better support for opening the most appropriate replica of Lotus Notes databases.
- ▶ **Easier-to-use, more accessible interface**—The new K-map user interface consists of two main tabs, Browse and Search, and Search Results. In addition, a new URL-addressable page appears for all the following actions:
 - Browse to a new category
 - Perform a search
 - Navigate to a different tab within the same set of search results
 - Perform a refined search
 - Navigate between sets of iterated search results
 - Page through a set of list results
 - Sort a list

Each page that appears as a result of these actions is put on the browser history, so you can return to it via the Back/Forward buttons in the browser interface. You can also bookmark these pages.

Lotus is also committed to offering an accessible interface for Discovery Server. This includes supporting all functionality with consistent and simple keyboard navigation of Discovery Server controls, fields, and hyperlinks. We will also aid low-vision users by:

- Providing enhanced color contrast in the standard interface design

- Supporting the Windows “High Contrast” palette schemes in the Appearance tab of the Control Panel - Display settings
- Supporting user-defined font settings in both the operating system and the browser

Other K-map user interface features introduced in 2.0 include:

- K-map search results offer People summaries in search results. We've also added a “Go to K-map Category” link to the summary of each document in search results.
 - Supported documents types include documents stored in Exchange public folders. Discovery Server 2.0 also offers better handling of Lotus Notes attachments, OLE embeddings, Domino.Doc documents and Lotus Team Workplace documents.
 - Bookmarking lets you bookmark (whether from the Search or Browse page) and have Discovery Server save both the Browse state and the Search parameters.
 - Saving state maintains a single saved state so that whenever you access the K-map from a particular computer, the interface is configured exactly as it was the last time you logged in. In response to customer feedback, the K-map no longer returns you to the category you were in when you last closed the K-map. This helps improve performance. Key information saved through this feature includes:
 - Document and People List heights in Browse and Search
 - Column order, in Document and People Lists in Browse and Search; and on Document, People, and Category tabs in Search Results
 - Sort order in Document and People Lists in Browse and Search
 - Column widths, in Document and People Lists in Browse and Search; and on Document, People, and Category tabs in Search Results
 - Summaries on/off, in Document and People Lists in Browse and Search; and on Document, People, and Category tabs in Search Results
 - ▶ The **K-map Editor** interface now lets you display a special K-map Editor accessible Reports view. This lets you create, schedule, and delete two new reports to aid your editing. These reports show both the number of documents per category, subtotalized per branch, and the documents that are new to a category (having been added automatically or manually).
- Another new K-map Editor feature is category visibility. This option allows you to keep categories hidden until they are ready to be viewed by your users. Then when you publish a category, it becomes visible to end users via the K-map interface, as long as its parent categories are also published. (New

categories assume the visibility of their parent by default.) Note that hiding a category does not prevent users from seeing the documents in that category if they are returned via K-map search. However, it does prevent you from getting affinities to that category.

To make it easier to optimize taxonomies, Discovery Server 2.0 offers two new options, Request Subdivide and Request Retrain.

Request Subdivide tells the K-map Building Service to divide the selected category into subcategories. When you select this option, the K-map Building Service attempts to create these subcategories, based on the total number of documents in the category (and the maximum number of documents per category you specified). If the K-map Building Service determines it can create two or more subcategories with reasonable fit values (and fairly evenly distributed documents), it moves documents from the specified category to the new subcategories, leaving the original category empty. If the selected category can't be usefully subcategorized, the K-map Building Service returns an error message.

Request Retrain helps “teach” the K-map Building Service how to categorize documents the way you want. After you manually move documents into a category, this option tells the K-map Building Service to place new documents with similar content into this category in the future. You can retrain selected categories, or the entire K-map taxonomy.

Other K-map Editor enhancements include an icon that indicates whether documents will launch in their native application or in the browser, a set Document Status options to set status for selected documents to locked or unlocked, and a Doc Counts option that displays the number of documents per category

- ▶ In response to user demand, we've made the **K-map Building Service** smarter when handling categorization. For example, in Discovery Server 1.0, if you manually moved a document into a category, K-map Building Service automatically assumed you wanted it to remain there forever, even if the category grew so big it required subcategorization to navigate properly. It would never change the categorization of this document. In Discovery Server 2.0, the K-map Building Service is free to subcategorize these documents as appropriate.

By default, all categories created by the K-map Building Service are hidden, so you can review them prior to making the categories available to users.

You can also import an existing file system taxonomy and use it as the basis for your K-map. To do this, Discovery Server 2.0 lets you import the file/folder document taxonomy from your operating system. You can then use the K-map Editor to modify the taxonomy into the K-map you want.

- ▶ **Profiles** include enhanced people awareness, the ability to determine the online status of selected members of your organization. We've also upgraded Person profile documents, which now include a more accessible interface.

- **People awareness** incorporates Lotus Instant Messaging functionality to transform passive name references into dynamic resources. These provide information about the person's current online status. Directly from the name reference, you can contact the person, find out more about them, and initiate other application-specific commands.

When you log on Discovery Server 2.0 (either via the K-map or through a profile), you automatically log on to the Lotus Instant Messaging server. Then anywhere your name appears within Discovery Server, other users will be able to tell you are online. Additionally, wherever they see your name displayed in K-map or in a profile, a Lotus Instant Messaging status icon appears to indicate your online status. (If you don't have a Lotus Instant Messaging server specified in your Domino Directory Person document, Discovery Server assumes Lotus Instant Messaging is not available and doesn't display your online status.)

People awareness lets you identify your most knowledgeable people in a particular area, and then contact them immediately by initiating a Lotus Instant Messaging dialog. This is especially useful when you need immediate information or need a question answered in real time.

- **Person profiles**—We've given Person profile documents a new servlet-based interface, to ensure the same "look-and-feel" as K-map. This also gives you better support for text resizing using your browser's *View - Text Size* commands. We've also made the Person profile interface more accessible.

One Discovery Server customer request is the ability to see all names the user is known by throughout the system. So at the bottom of the Contact Information page, we've added a new field called "Other user names."

We've also made some changes to the Affinities interface. We've moved the interface for approving proposed affinities into the profile document. For example, if there are proposed affinities, they appear in a table within the profile document. (Note that if you approve proposed affinities but then cancel out of the profile document without saving it, the approvals are also cancelled.)

And in response to other user feedback, we display the "Declaring Affinities" description in Read Mode on the Affinities page if the person viewing the profile document has edit rights.

- ▶ **Spiders**—*Better support for Domino.Doc and Lotus Team Workplace*

Discovery Server 2.0 supports spidering Domino.Doc and Lotus Team Workplace documents. We've also enhanced other spiders, such as the one for Lotus Notes/Domino, and added a spider for Exchange.

- **Domino.Doc spider**—We offer spidering for both Domino.Doc 3.1 and 3.0. Capabilities include:
 - Documents and their attachments are treated as single documents.
 - A Domino.Doc repository is defined at the File Cabinet level.
 - Forum (discussion) docs are spidered.
 - There is an administration option to spider all, or only latest versions.
 - Domino.Doc-specific field mapping is spidered.
 - There is support for archived documents.
- **Lotus Team Workplace spider**—We also support spidering for Lotus Team Workplace. Capabilities include:
 - A Lotus Team Workplace repository defined at the Main.nsf level.
 - Embedded pages are treated as one entity.
 - Lotus Team Workplace-specific field mapping is spidered.
- **Handling documents with attachments**—Spiders in Discovery Server 2.0 have improved the handling of documents that do not contain text but do contain one or more attachments. These so-called “sparse container” documents will be classified with their attachments. The title of the attachment identifies the sparse container document.
- ▶ **Metrics**—More detailed and easier to understand

Metrics now consists of four separate services: Profile Maintenance, Metrics Reporting, Affinity Processing, and Metrics Processing. These comprised a single service in Discovery Server 1.0; we separated them in response to user demand for a less “black box” approach to Metrics. All four services can only run on one server at a time; but you can move them from one server to another, and schedule them individually.

 - Profile Maintenance processes user edits to profile documents.
 - Metrics Reporting creates Metrics reports.
 - Metrics Processing computes document values, updates the Discovery Server data with new affinities and new document values, and updates full-text search with new values.
 - Affinity Processing calculates affinities, proposes and publishes affinities, sends affinity e-mails, and updates profile documents with affinities. This service provides more control over when affinities are generated.
- ▶ **Installation, setup, and maintenance**—More customer control

Discovery Server 2.0 offers significantly enhanced administration functionality. This lets you have greater control over installation, setup, and

maintenance. And you'll have a better idea of what's going on "under the hood" of Discovery Server.

- **Installation**—The installation dialog box now includes numerous checks and warnings to better guide you through installing and upgrading. Also, we no longer install the K-map Editor with the server. We have discovered that in practice, installing the K-map Editor on the server is rarely done—usually only for demo purposes.

We also modified the Admin Name & Password screen to let you provide your own user name. This can be an account already defined on the system or created on the fly if one doesn't exist.

- **Setup**—In Release 2.0, we re-worked the setup screens to be easier and more intuitive. This includes better input validation and error messages.
- **Maintenance**—As a response to customer requests for more control over Discovery Server (and to accommodate new features introduced in Release 2.0), we have incorporated new administration functionality. For example, we made the interface for enabling the XML spider always visible, instead of requiring you to set an INI variable to do this.

We've also added a new interface for replicating K-map data to a secondary server. This includes a "K-map Replication" checkbox in the Server document on the primary Discovery Server, as well as a "K-map Replica" checkbox in the Server document of Secondary Discovery Servers. After you check this option and save the Server document, replication will copy the K-map data from the primary to the specified secondary servers. These secondaries can then serve end users, helping you balance user workload among several machines.

Other maintenance features include:

- A new interface for enabling the Exchange spider
- A drill-down model to support large numbers of repositories
- Better "paging" capability in multi-page views and logs
- The ability to move Metrics processing from one server to another
- Improved interface for choosing repositories for creating the K-map
- Updates to Service Status views
- **Logging**—We've improved logging messages to be more informative. For example, the K-map Building log has been significantly enhanced, with more logging of editor activities. And to keep you better apprised as Lotus Team Workplace and Domino.Doc spiders run, we provide a new message type to enable these spiders to post interim begin/end messages as each room/binder is processed.

Other enhancements

- ▶ **Discovery Server 2.0 API Toolkit**—As mentioned earlier, users have asked for a more open approach. To meet this need, we're providing a complete Discovery Server API Toolkit. This allows you to customize Discovery Server to suit your organization's exact requirements. Third-party software developers will also use this Toolkit to develop their own solutions based on Discovery Server functionality.
- ▶ **Data repositories**—Discovery Server 2.0 includes many new features to better manage your data repositories. For example, you can temporarily prevent a repository from being spidered, and have it start up again later, without disabling the spider. You can then have the repository requeued automatically. You can also stop spidering a repository (for instance, because you made a mistake in defining it), delete it, and start over with new parameters. Other new data repository functionality include:
 - Preventing two repository records from pointing to the same source
 - Supporting a “delete and make new copy” action from the main Repositories view
 - Providing the ability to unqueue a repository
 - An interface for spidering Exchange e-mail and Public Folders
 - Options for traversing Domino.Doc and Lotus Team Workplace hierarchy
 - Options for spidering Domino.Doc and Lotus Team Workplace revisions
 - The ability to edit Field Map fields after the repository has been queued/spidered

And we've added an interface for specifying a subset of repository data to be spidered, per customer request.

- ▶ **Field mapping**—Our goal in this release is to only map what the administrator selects to map—in other words, what you define in the profile forms. This means there will be no “identity mapping” (automatically mapping a field name in the source to the same field name in the profile document). Also in Release 2.0, we provide better coverage for well-known data mappings in the default data field map (\$Global).
- ▶ **People sources**—New features in this area include:
 - An interface to specify “Process all documents during next run”
 - An interface to associate supplemental sources with all or some subset of authoritative sources
 - Support for additional LDAP search parameters BASEDN and SCOPE

- ▶ **Documentation**—Discovery Server 2.0 documentation includes a Deployment Guide, available shortly after product ship. This should be a “must-read” before a customer site begins to install and implement Discovery Server.

Products

We recommend that you start with the following products:

- ▶ **IBM WebSphere Studio** is an open comprehensive development environment for building, testing and deploying dynamic on demand e-business applications. Founded on open technologies and built on Eclipse, WebSphere Studio provides a flexible, portal-like integration of multi-language, multi-platform and multi-device application development tools that maximize your productivity, increase ROI and improve overall time to value.
- ▶ **IBM WebSphere Application Server** is a high performance and extremely scalable transaction engine for dynamic e-business applications. The Open Services Infrastructure allows companies to deploy a core operating environment that works as a reliable foundation capable of handling high volume secure transactions and Web services. WebSphere continues the evolution to a single Web services-enabled, Java 2 Enterprise Edition (J2EE) application server and development environment that addresses the essential elements needed for an on demand operating environment.
- ▶ You can use **WebSphere Business Integration Server V4.2** to quickly integrate new or existing applications or systems on diverse platforms, create and rapidly deploy new business processes, or solve a variety of business integration needs.
- ▶ **DB2 Universal Database**—DB2 Version 8.1 helps solve critical business problems by integrating information across the entire enterprise by leveraging federated Web services and XML. DB2 is delivering new federated capabilities that enable customers to integrate information as Web services. DB2 also delivers new XML enhancements that make it easier for programmers to integrate DB2 and XML information.
- ▶ **DB2 Information Integrator**—This new family of products is designed to help you integrate structured, semi-structured and unstructured information effectively and efficiently. These products, based on the previously disclosed Xperanto project, provide the foundation for a strategic information integration framework to help access, manipulate, and integrate diverse, distributed and real time data.
- ▶ **Lotus Domino 6** represents an evolution of the platform, adding features that increase the strength of Domino as a strategic platform for Web applications, messaging, and e-business. New enhancements for application design and expanded support for standards-based development methodologies give

developers of all skill levels even more capabilities to add collaboration to new or existing Web, Lotus Notes, or mobile applications. Domino 6 server provides platform choice in both hardware and operating system, with further enhancements in security, administration, performance, enterprise data integration, and directory options.

- ▶ **Rational Rose family**—The Unified Modeling Language (UML) has become the software industry's standard notation for representing software architecture and design models. Many development organizations are finding that modeling with the UML helps them build better software faster. Rational Rose software is your solution for building better software faster with the UML.
- ▶ **Lotus Discovery Server**—This knowledge management server for e-business users provides organizations with the easiest way to organize and locate resources across various systems and information sources.
- ▶ **Lotus Instant Messaging** is real time collaboration software with instant messaging, whiteboarding, and application sharing capabilities.
- ▶ **Tivoli Identity Manager**—Companies are increasing the number of users (customers, employees, partners and suppliers) who are allowed to access information. As IT is challenged to do more with fewer resources, effectively managing user identities throughout their lifecycle is even more important. IBM Tivoli Identity Manager provides policy-based identity management across legacy and e-business environments. Intuitive Web administrative and self-service interfaces integrate with existing business processes to help simplify and automate managing and provisioning users. It incorporates a workflow engine and leverages identity data for activities such as audit and reporting.

A.2.3 Virtualization

Virtualized networks serve up computing as needed. Emerging grid technologies, for example, allow a collection of resources to be shared and managed just as if they were one large, virtualized computer.

IBM Federated Database technology

Note: The following is from the article of the same name by Laura Haas and Eileen Lin (Information Integration Development, San Jose), published March 2002. IBMers can find this material at
<http://www7b.software.ibm.com/dmdd/library/techarticle/0203haas/0203haas.html>.

In a large modern enterprise, it is almost inevitable that different portions of the organization will use different database management systems to store and

search their critical data. Competition, evolving technology, mergers, acquisitions, geographic distribution, and the inevitable decentralization of growth all contribute to this diversity. Yet it is only by combining the information from these systems that the enterprise can realize the full value of the data they contain.

For example, in the finance industry, mergers are an almost commonplace occurrence. The newly created entity inherits the data stores of the original institutions. Many of those stores will be relational database management systems, but often from different manufacturers; for instance, one company may have used primarily Sybase, and another Informix® IDS. They may both have one or more document management systems—such as Documentum or IBM Content Manager—for storing text documents such as copies of loans, etc. Each may have applications that compute important information (for example, the risk of a loan to a given customer), or mine for information about customers' buying patterns.

After the merger, they need to be able to access all customer information from both sets of stores, analyze their new portfolios using existing and new applications, and, in general, use the combined resources of both institutions through a common interface. They need to be able to identify common customers and consolidate their accounts, although the different companies may have referred to their customers using totally different identifying keys. Federation technologies can significantly ease the pain in these situations by providing a unified interface to diverse data.

IBM has made significant investments in federation technologies that have resulted in market leading capabilities across the Data Management product portfolio. Today, federation capabilities enable unified access to any digital information, in any format—structured and unstructured, in any information store. Federation capabilities are available today through a variety of IBM products including DB2 UDB (and DB2 Relational Connect), DB2 DataJoiner®, and IBM Enterprise Information Portal (EIP). This set of federation technologies continues to be enhanced and our customers' investments in all of these products continue to deliver real business value.

This section focuses specifically on advanced database federation capabilities, implemented through a technology sometimes referred to by the code name *Garlic*, which represent the next generation of information federation enhancements from IBM software. These enhancements will enable clients to access and integrate the data and specialized computational capabilities of a wide range of relational and non-relational data sources. The Garlic technology will be incorporated into all IBM software offerings that provide federation technology over time. Customers may rest assured that not only will their investments in existing products be protected, but also that in the future, no

matter which product is selected, they will be able to leverage the advanced capabilities described here.

IBM's federated database systems offer powerful facilities for combining information from multiple data sources. Built on best-of-breed technology from an earlier product, DB2 DataJoiner, and enhanced with leading-edge features for extensibility and performance from the Garlic research project, IBM's federated database capabilities are unique in the industry. DB2 DataJoiner introduced the concept of a virtual database, created by federating together multiple heterogeneous relational data sources. Users of DB2 DataJoiner could pose arbitrary queries over data stored anywhere in the federated system, without worrying about the data's location, the SQL dialect of the actual data stores, or the capabilities of those stores. Instead, users had the full capabilities of DB2 against any data in the federation. The Garlic project demonstrated the feasibility of extending this idea to build a federated database system that effectively exploits the query capabilities of diverse, possibly non-relational data sources. In both of these systems, as in today's DB2, a middleware query processor develops optimized execution plans and compensates for any functionality that the data sources may lack.

In this section, we describe the key characteristics of IBM's federated technology: transparency, heterogeneity, a high degree of function, autonomy for the underlying federated sources, extensibility, openness, and optimized performance. We then "roll back the covers" to show how IBM's database federation capabilities work. We illustrate how the federated capabilities can be used in a variety of scenarios, and conclude with some directions for the future.

Characteristics of the IBM federated solution

- ▶ **Transparency**—If a federated system is transparent, it masks from the user the differences, idiosyncrasies, and implementations of the underlying data sources. Ideally, it makes the set of federated sources look to the user like a single system. The user should not need to be aware of where the data is stored (location transparency), what language or programming interface is supported by the data source (invocation transparency), if SQL is used, what dialect of SQL the source supports (dialect transparency), how the data is physically stored, or whether it is partitioned and/or replicated (physical data independence, fragmentation and replication transparency), or what networking protocols are used (network transparency). The user should see a single uniform interface, complete with a single set of error codes (error code transparency). IBM provides all these features, allowing applications to be written as if all the data were in a single database, although, in fact, the data may be stored in a heterogeneous collection of data sources.
- ▶ **Heterogeneity** is the degree of differentiation in the various data sources. Sources can differ in many ways. They may run on different hardware, use

different network protocols, and have different software to manage their data stores. They may have different query languages, different query capabilities, and even different data models. They may handle errors differently, or provide different transaction semantics. They may be as much alike as two Oracle instances, one running Oracle 8i, and the other Oracle 9i, with the same or different schemas. Or they may be as diverse as a high-powered relational database, a simple, structured flat file, a Web site that takes queries in the form of URLs and spits back semi-structured XML according to some DTD, a Web service, and an application that responds to a particular set of function calls. IBM's federated database can accommodate all of these differences, encompassing systems such as these in a seamless, transparent federation.

- ▶ **A high degree of function**—IBM's federated capability provides users with the best of both worlds: all the function of its rich, standard-compliant DB2 SQL capability against all the data in the federation, as well as all the function of the underlying data sources. DB2's SQL includes support for many complex query features, including inner and outer joins, nested subqueries and table expressions, recursion, user-defined functions, aggregation, statistical analyses, automatic summary tables, and others too numerous to mention. Many data sources may not provide all of these features. However, users still get the full power of DB2 SQL on these sources' data, because of function compensation. Function compensation means that if a data source cannot do a particular query function, the federated database retrieves the necessary data and applies the function itself. For example, a file system typically cannot do arbitrary sorts. However, users can still request that data from that source (in other words, some subset of a file) be retrieved in some order, or ask that duplicates be eliminated from that data. The federated database will simply retrieve the relevant data, and do the sort itself.

While many sources do not provide all the function of DB2 SQL, it is also true that many sources have specialized functionality that the IBM federated database lacks. For example, document management systems often have scoring functions that let them estimate the relevancy of retrieved documents to a user's search. In the financial industry, time series data is especially important, and systems exist that can compare, plot, analyze, and subset time series data in specialized ways. In the pharmaceutical industry, new drugs are based on existing compounds with particular properties. Special-purpose systems can compare chemical structures, or simulate the binding of two molecules. While such functions could be implemented directly, it is often more efficient and cost-effective to exploit the functionality that already exists in data sources and application systems. IBM allows the user to identify functions of interest from the federated sources, and then to use them in queries, so that no function of a source need be lost to the user of the federated system.

- ▶ **Extensibility and openness of the federation**—All systems need to evolve over time. In a federated system, new sources may be needed to meet the changing needs of the users' business. IBM makes it easy to add new sources. The federated database engine accesses sources via a software component known as a *wrapper*. Accessing a new type of data source is done by acquiring or creating a wrapper for that source. The wrapper architecture enables the creation of new wrappers. Once a wrapper exists, simple data definition (DDL) statements allow sources to be dynamically added to the federation without stopping ongoing queries or transactions.

Any data source can be wrapped. IBM supports the ANSI SQL/MED standard (MED stands for Management of External Data). This standard documents the protocols used by a federated server to communicate with external data sources. Any wrapper written to the SQL/MED interface can be used with IBM's federated database. Thus wrappers can be written by third parties as well as IBM, and used in conjunction with IBM's federated database.

- ▶ **Autonomy for data sources**—Typically a data source has existing applications and users. It is important, therefore, that the operation of the source is not affected when it is brought into a federation. IBM's federated database does not disturb the local operation of an existing data source. Existing applications will run unchanged, data is neither moved nor modified, interfaces remain the same. The way the data source processes requests for data is not affected by the execution of global queries against the federated system, though those global queries may touch many different data sources. Likewise, there is no impact on the consistency of the local system when a data source enters or leaves a federation. The sole exception is during federated two phase commit processing for sources that participate. Data sources involved in the same unit of work will need to participate in commit processing and can be requested to roll back the associated changes if necessary.

Unlike other products, our wrapper architecture does not require any software to be installed on the machine that hosts the data source. We communicate with the data source via a client server architecture, using the source's normal client. In this way, IBM's federated data source looks like just another application to the source.

- ▶ **Optimized performance**—The optimizer is the component of a relational database management system that determines the best way to execute each query. Relational queries are non-procedural and there are typically several different implementations of each relational operator and many possible ordering of operators to choose from in executing a query. While some optimizers use heuristic rules to choose an execution strategy, IBM's federated database considers the various possible strategies, modeling the

likely cost of each, and choosing the one with the least cost. (Typically, cost is measured in terms of system resources consumed).

In a federated system, the optimizer must decide whether the different operations involved in a query should be done by the federated server or by the source where the data is stored. It must also determine the order of the operations, and what implementations to use to do local portions of the query. To make these decisions, the optimizer must have some way of knowing what each data source can do, and how much it costs. For example, if the data source is a file, it would not make sense to assume it was smart, and ask it to perform a sort or to apply some function. On the other hand, if the source is a relational database system capable of applying predicates and doing joins, it might be a good idea to take advantage of its power if it will reduce the amount of data that needs to be brought back to the federated engine. This will typically depend on the details of the individual query. The IBM optimizer works with the wrappers for the different sources involved in a query to evaluate the possibilities. Often the difference between a good and a bad decision on the execution strategy is several orders of magnitude in performance. IBM's federated database is unique in the industry in its ability to work with wrappers to model the costs of federated queries over diverse sources. As a result, users can expect the best performance possible from their federated system.

To further enhance performance, each wrapper implementation takes advantage of the operational knobs provided by each data source using the source's native API. For example, blocking multiple result rows into one message (also known as *block fetch*) is a common performance knob. The query compiler will communicate with the wrapper to indicate which query fragments can utilize block fetch and thus achieve the maximal performance at runtime without loss of query semantics.

How IBM's federated capabilities work

- ▶ **Architecture**—IBM's federated database architecture is shown in Figure A-23 on page 327. Applications can use any supported interface (including ODBC, JDBC, or a Web service client) to interact with the federated server. The federated server communicates with the data sources by means of software modules called wrappers.

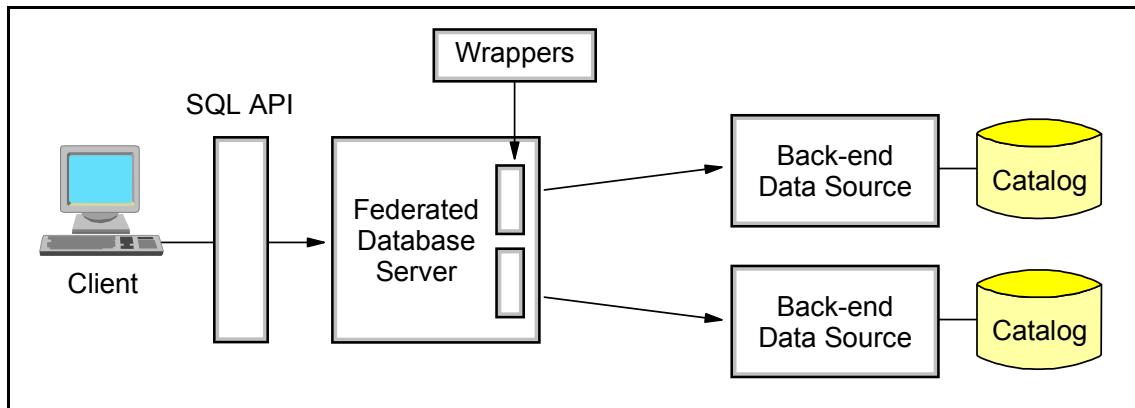


Figure A-23 Architecture of an IBM federated system

- ▶ **Configuring a federated system**—A federated system is created by installing the federated engine and then configuring it to talk to the data sources.
- ▶ **Query processing**—After the federated system is configured, an application can submit a query written in SQL to a federated server. The federated server optimizes the query, developing an execution plan in which the query has been decomposed into fragments that can be executed at individual data sources. As mentioned above, many decompositions of the query are possible, and the optimizer chooses among alternatives on the basis of minimum estimated total resource consumption. Once a plan has been selected, the federated database drives the execution, invoking the wrappers to execute the fragments assigned to them. To execute a fragment, the wrapper performs whatever data source operations are needed to carry it out, perhaps a series of function calls or a query submitted to the data source in its native query language. The resulting streams of data are returned to the federated server, which combines them, performs any additional processing that could not be accomplished by a data source, and returns the final result to the application.

At the heart of IBM's approach to federated query processing is the manner in which the federated server's optimizer and the wrappers together arrive at a plan for executing the query. The optimizer is responsible for exploring the space of possible query plans. Dynamic programming is the default method used in join enumeration, with the optimizer first generating plans for single-table accesses, then for two-way joins, etc. At each level, the optimizer considers various join orders and join methods, and if all the tables are located at a common data source, it considers performing the join either at the data source or at the federated server. This process is illustrated in Figure A-24.

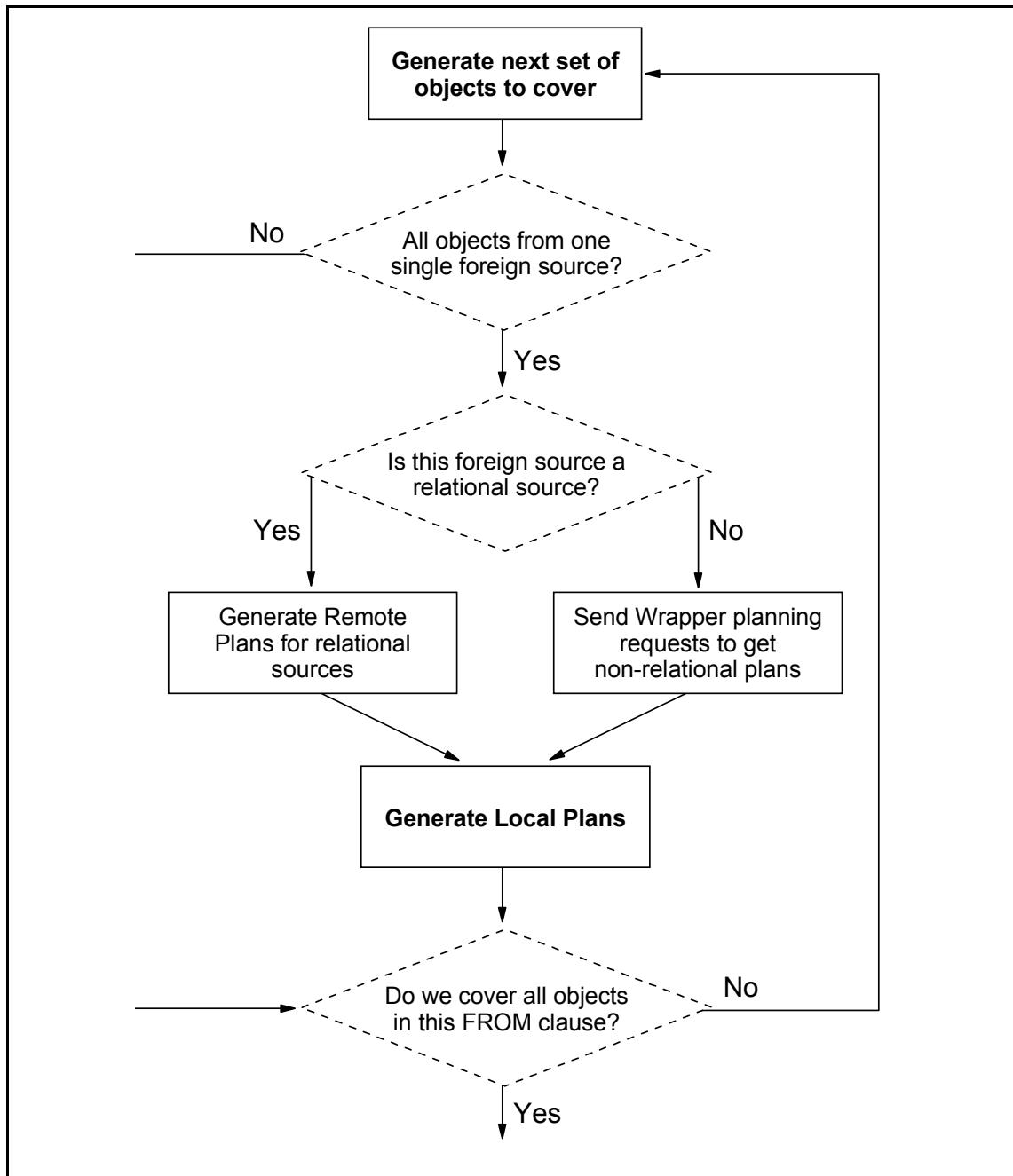


Figure A-24 Query planning for joins

The optimizer works differently with relational and non-relational wrappers. The optimizer models relational sources in detail, using information provided by the wrapper to generate plans that represent what it expects the source to do.

However, because non-relational sources do not have a common set of operations or common data model, a more flexible arrangement is required with these sources. Hence the optimizer works with the non-relational wrappers:

- The IBM federated database submits candidate query fragments called “requests” to a wrapper if the query fragments apply to a single source.
- When a non-relational wrapper receives a request, it determines what portion, if any, of the corresponding query fragment can be performed by the data source.
- The wrapper returns a reply that describes the accepted portion of the fragment. The reply also includes an estimate of the number of rows that will be produced, an estimate of the total execution time, and a wrapper plan: an encapsulated representation of everything the wrapper will need to know to execute the accepted portion of the fragment.
- The federated database optimizer incorporates the reply into a global plan, introducing additional operators as necessary to compensate for portions of fragments that were not accepted by a wrapper. The cost and cardinality information from the replies is used to estimate the total cost of the plan, and the plan with minimum total cost is selected from among all the candidates. When a plan is selected, it need not be executed immediately; it can be stored in the database catalogs and subsequently used one or more times to execute the query. Even if a plan is used immediately, it need not be executed in the same process in which it was created, as illustrated in Figure A-25 on page 330.

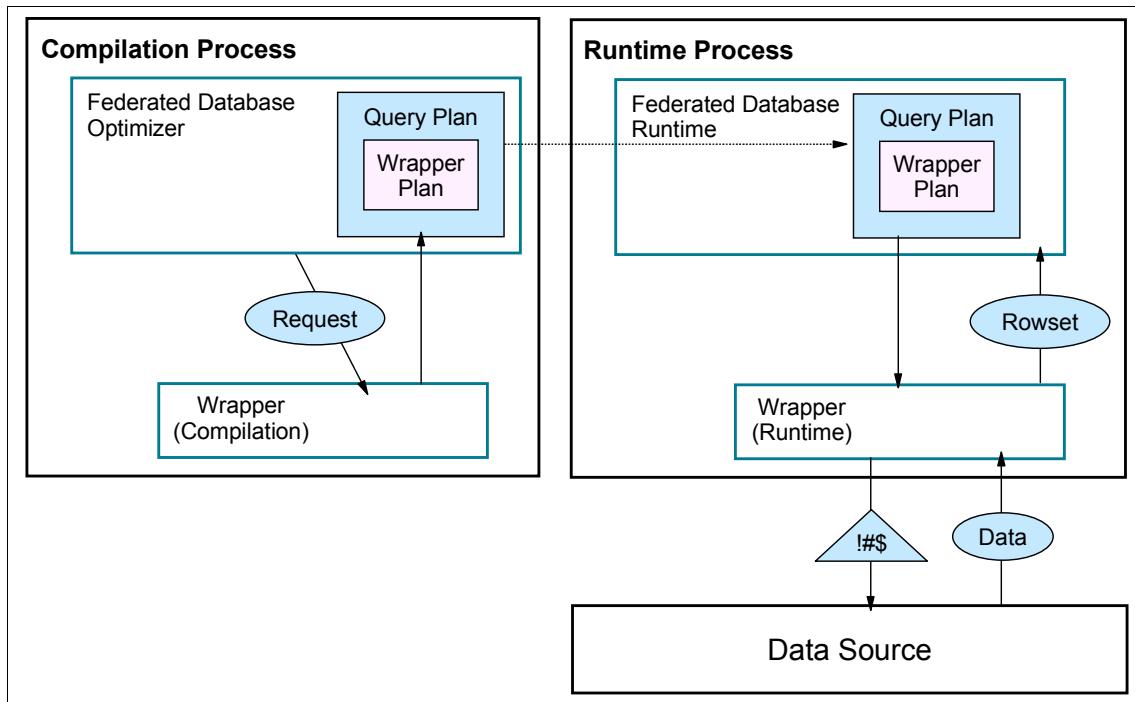


Figure A-25 Compilation and runtime for non-relational sources

Using a federated database system

Why is a federated system useful? How do customers use federation capabilities? In general, a federated system is useful in any situation in which there are multiple sources of data, and a need to combine the information from these various sources. In this section we look at how some customers are using IBM's federated technology to solve their business problems today.

- ▶ **Distributed operations: A major pharmaceutical company**—Many companies today are global companies, with a need to coordinate activities in multiple locations throughout the world. For example, a pharmaceutical company might have research labs in both Europe and the US. Each of the labs houses scientists looking for new drugs to battle particular diseases. The scientists all have access to databases of chemical compounds, stored in special-purpose systems that allow searching by particular characteristics of the compounds or by chemical structure (structural similarity). In both labs, the scientists run high throughput screenings of compounds to test their effectiveness against different biological targets. The results of these tests are stored in relational databases at each lab. Other data sources accessed by the scientists include large flat files of genomic and proteomic information,

patent databases, spreadsheets of data and analysis, images and text documents.

The scientists in the two labs have different missions, different cures or treatments that they are pursuing. This leads them to do different experiments, and to focus on particular sets of compounds. However, often the same compounds may be useful against different targets, and sometimes one test may be a good indicator of results for other tests. Thus it is important for the scientists at one lab to have access to the data being produced at the other, so as not to duplicate effort. While this could be accomplished by building a large warehouse with all the compound data and test results, there are several drawbacks to that approach. First, the test result data changes rapidly, with thousands of records being added every day from both sides of the Atlantic, making maintenance difficult. Second, the warehouse must either be replicated at both sites, or one site must suffer slower performance for accessing the data. Replication adds to the cost of the solution and the complexity of maintenance. Third, the compound data, today stored in specialized repositories, would need to be migrated to a relational base, including re-implementing the search algorithms and any existing applications.

A federated solution eliminates these issues. Data is left in the existing data sources, with their native access paths, and current applications run unchanged. However, it is easy to build new applications that can access data from any of the sources, regardless of continent. Local data stays local, for rapid access. The less frequently used remote data is still accessible, as needed, and queries are optimized by the federated server to ensure that they are retrieved as efficiently as possible. Replication can still be used if desired for those portions of the data that are heavily accessed by both laboratories.

- ▶ **Heterogeneous replication**—Many businesses choose to keep multiple copies of their data. For example, one major retailer with outlets all over the United States backs up data from its various locations at regional warehouses. The retail outlets use one relational database management system; the warehouses are implemented using another DBMS that scales better. However, this poses the problem of how to transfer the data from source to warehouse. IBM's federated technology makes it easy to not only move data, selecting it from the sources and inserting it into the warehouse, but to re-shape it as well, aggregating information from the various outlets before inserting it into the warehouse.

IBM provides a replication product, DB2 DataPropagator™ that helps you integrate your distributed database environment by replicating data between relational databases using the features of the federated database.

DataPropagator automates the copying of data between remote systems, avoiding the need to unload and load databases manually. For a non-DB2

relational source, Capture triggers are defined to capture changes to the source and write them to a staging table. The Apply program, which runs on an IBM federated database server, uses a nickname for the staging table to copy the changes from the staging table to a target table in the IBM federated database or in another non-DB2 relational database. Heterogeneous replication is thus made easy thanks to federated technology.

- ▶ **Distributed data warehouse**—Implementing a distributed data warehouse has been shown to provide higher availability and lower overall cost. An enterprise can create several data marts that store only high level summaries of data derived from the warehouse. With IBM's federated technology, data marts and warehouse can be on separate systems, yet users of the data mart can still drill down with ease from their local level of summarization into the warehouse. Federated technology shields the users, who have no need to know that the data warehouse is distributed, by providing a virtual data warehouse.
- ▶ **Geospatial application**—A bank needs to choose a location for a new branch office. The location chosen must maximize the expected profit. To do so, the bank needs to consider for each location the demographics of the surrounding neighborhood (Do the demographics fit the targeted customer base?), the crime rate in the area (A low crime rate is important for retail operations.), proximity of the site to major highways (to attract customers from outside the immediate area), proximity of major competitors (A site with little competition will most likely mean higher sales.), and proximity to any known problem areas that must be avoided (A dump or other unattractive feature of the neighborhood could negatively impact business.). Some of the needed information will come from the bank's own databases. Other information must be retrieved from external data stores with information on the community. This application illustrates the need to integrate geospatial data with traditional business data. It requires advanced query analysis functions for correlating the data, and end user tools that can visually display the data in a geospatial context.

Traditionally, geospatial data have been managed by specialized geographic information systems (GISes) that cannot integrate spatial data with other business data stored in the company's RDBMS and in external data sources. DB2 Spatial Extender is the product of collaboration with an IBM partner, the Environmental Systems Research Institute (ESRI). DB2 Spatial Extender works with an IBM federated database to give customers the best of both worlds. The customer can take advantage of the geospatial intelligence built into the DB2 Spatial Extender combined with the vast amount of available business information from the federated system. This enables the organization to enhance its understanding of its business, leverage the value of existing data, and build sophisticated new applications, leading to business success.

Conclusions

Despite considerable attention from the research community, few commercial database management systems have addressed the problem of integrating relational and non-relational data sources into a federation. With its federation capabilities, IBM has made significant progress toward this goal. IBM's unique federated query processing technology allows users to enjoy all the power of DB2 SQL coupled with the power of individual data sources. It provides users with all the benefits of transparency, heterogeneity, a high degree of function, autonomy for the underlying federated sources, extensibility, openness, and optimized performance. Federation is being used today to solve many important business needs.

In the future, we will continue to work to improve the performance and functionality of the federation. For example, a form of caching can already be accomplished using an automatic summary table (AST) mechanism, which allows administrators to define materialized views of data in a set of underlying tables—or nicknames. For certain classes of queries, the database can automatically determine whether a query can be answered using an AST, without accessing the base tables. In addition to constantly improving performance, we are also working on tools to aid in the configuration, tuning and administration of federated systems. Tools for generating statistics for data from non-relational sources and for monitoring the behavior of a federated system are underway. Tools to assist wrapper developers are also in development.

Finally, even a well-designed federated database management system and an accompanying set of tools remains a partial solution to the larger problem of data integration. A comprehensive solution will have to integrate applications as well as data, and address higher-level issues like data quality, annotation, differences in terminology, and business rules that indicate when and how information is to be combined. IBM is focusing on this broader set of information integration requirements to enable customers to satisfy their business integration requirements, and database-style federation is just one key integration technology.

From data management to information integration: A natural evolution

Note: The following is from the article of the same name by Mary Roth and Dan Wolfson (DBTI for e-business, IBM Silicon Valley Lab), published in June 2002. IBMers can find this at
<http://www7b.software.ibm.com/dmdd/library/techarticle/0206roth/0206roth.html>.

The explosion of the Internet and e-business in recent years has caused a secondary explosion of information. Industry analysts predict that more data will be generated in the next three years than in all of recorded history. Enterprise business applications can respond to this information overload in one of two ways: they can bend and break under the sheer volume and diversity of such data, or they can harness this information and transform it into a valuable asset by which to gain a competitive advantage in the marketplace.

Because the adoption of Internet-based business transaction models has significantly outpaced the development of tools and technologies to deal with the information explosion, many businesses find themselves unintentionally using the former approach. Significant development resources are spent on quick and dirty integration solutions that cobble together different data management systems (databases, content management systems, enterprise application systems) and transform data from one format to another (structured, XML, byte streams). Revenue is lost when applications suffer from scalability and availability problems. New business opportunities are simply overlooked because the critical nuggets of information required to make a business decision are lost among the masses of data being generated.

In this section, we propose a technology platform and tools to harness the information explosion and provide an end-to-end solution for transparently managing both the volume and diversity of data that is in the marketplace today. We call this technology information integration. IBM provides a family of data management products that enable a systematic approach to solve the information integration challenges that businesses face today. Many of these products and technologies are showcased in the Information Integration technology preview.

The foundation of the platform is a state-of-the art database architecture that seamlessly provides both relational and native XML as first class data models. We believe that database technology provides the strongest foundation for an information integration platform for three significant reasons:

1. DBMSs have proven to be hugely successful in managing the information explosion that occurred in traditional business applications over the past 30 years. DBMSs deal quite naturally with the storage, retrieval, transformation, scalability, reliability, and availability challenges associated with robust data management.
2. The database industry has shown that it can adapt quickly to accommodate the diversity of data and access patterns introduced by e-business applications over the past six years. For example, most enterprise-strength DBMSs have built-in object-relational support, XML capabilities, and support for federated access to external data sources.

3. There is a huge worldwide investment in DBMS technology today, including databases, supporting tools, application development environments, and skilled administrators and developers. A platform that exploits and enhances the DBMS architecture at all levels is in the best position to provide robust end-to-end information integration.

This section is organized as follows:

- We briefly review the evolution of the DBMS architecture.
- We provide a real-world scenario that illustrates the scope of the information integration problem and sketches out the requirements for a technology platform.
- We formally call out the requirements for a technology platform.
- We present a model for an information integration platform that satisfies these requirements and provides an end-to-end solution to the integration problem as the next evolutionary step of the DBMS architecture.

Evolution of DBMS technology

Figure A-26 on page 336 captures the evolution of relational database technology. Relational databases were born out of a need to store, manipulate and manage the integrity of large volumes of data. In the 1960s, network and hierarchical systems such as CODASYL and IMSTM were the state-of-the-art technology for automated banking, accounting, and order processing systems enabled by the introduction of commercial mainframe computers. While these systems provided a good basis for the early systems, their basic architecture mixed the physical manipulation of data with its logical manipulation. When the physical location of data changed, such as from one area of a disk to another, applications had to be updated to reference the new location.

A revolutionary paper by Codd in 1970 and its commercial implementations changed all that. Codd's relational model introduced the notion of data independence, which separated the physical representation of data from the logical representation presented to applications. Data could be moved from one part of the disk to another or stored in a different format without causing applications to be rewritten. Application developers were freed from the tedious physical details of data manipulation, and could focus instead on the logical manipulation of data in the context of their specific application.

Not only did the relational model ease the burden of application developers, but it also caused a paradigm shift in the data management industry. The separation between what and how data is retrieved provided an architecture by which the new database vendors could improve and innovate their products. SQL became the standard language for describing what data should be retrieved. New storage schemes, access strategies, and indexing algorithms were developed to speed

up how data was stored and retrieved from disk, and advances in concurrency control, logging, and recovery mechanisms further improved data integrity guarantees. Cost-based optimization techniques completed the transition from databases acting as an abstract data management layer to being high-performance, high-volume query processing engines.

As companies globalized and as their data quickly became distributed among their national and international offices, the boundaries of DBMS technology were tested again. Distributed systems such as *R** and *Tandem* showed that the basic DBMS architecture could easily be exploited to manage large volumes of distributed data. Distributed data led to the introduction of new parallel query processing techniques (PARA), demonstrating the scalability of the DBMS as a high-performance, high-volume query processing engine.

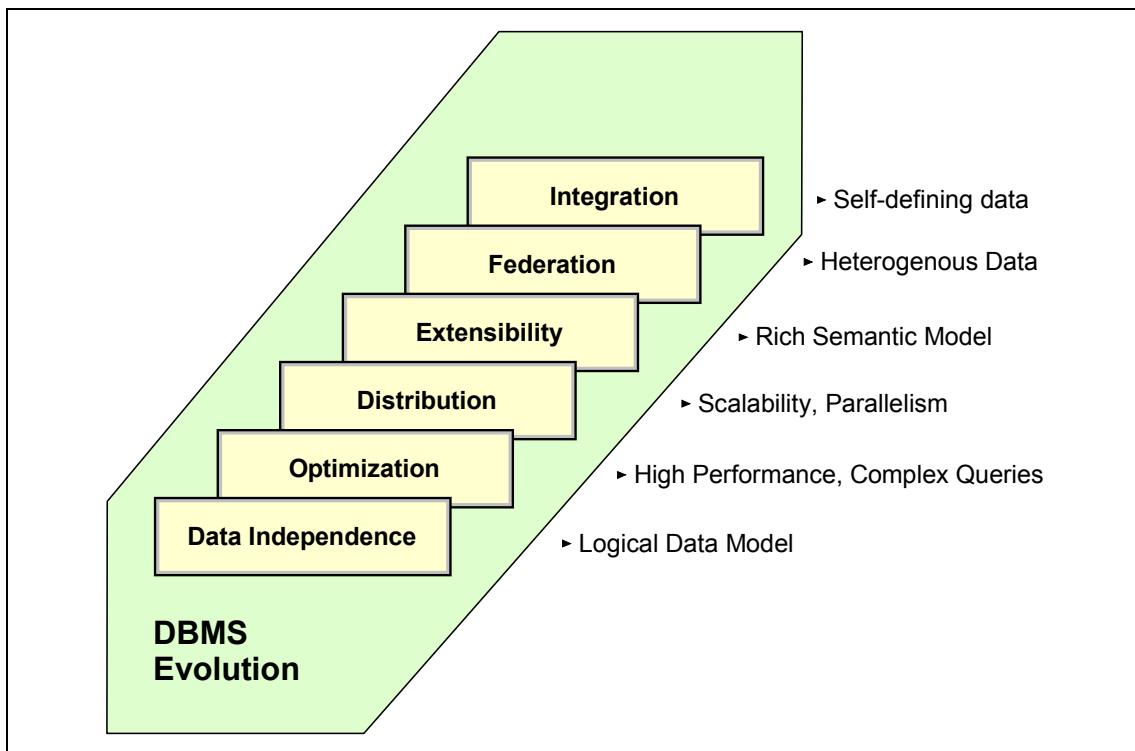


Figure A-26 Evolution of DBMS architecture

The lessons learned in extending the DBMS with distributed and parallel algorithms also led to advances in extensibility, whereby the monolithic DBMS architecture was replumbed with plug-and-play components (Starburst). Such an architecture enabled new abstract data types, access strategies and indexing schemes to be easily introduced as new business needs arose. Database

vendors later made these hooks publicly available to customers as Oracle data cartridges, Informix DataBlade™, and DB2 Extenders.

Throughout the 1980s, the database market matured and companies attempted to standardize on a single database vendor. However, the reality of doing business generally made such a strategy unrealistic. From independent departmental buying decision to mergers and acquisitions, the scenario of multiple database products and other management systems in a single IT shop became the norm rather than the exception. Businesses sought a way to streamline the administrative and development costs associated with such a heterogeneous environment, and the database industry responded with federation. Federated databases provided a powerful and flexible means for transparent access to heterogeneous, distributed data sources.

We are now in a new revolutionary period enabled by the Internet and fueled by the e-business explosion. Over the past six years, Java and XML have become the vehicles for portable code and portable data. To adapt, database vendors have been able to draw on earlier advances in database extensibility and abstract data types to quickly provide object-relational data models, mechanisms to store and retrieve relational data as XML documents (XTABLES), and XML extensions to SQL (SQLX).

The ease with which complex Internet-based applications can be developed and deployed has dramatically accelerated the pace of automating business processes. The premise of this section is that the challenge facing businesses today is information integration. Enterprise applications require interaction not only with databases, but also content management systems, data warehouses, workflow systems, and other enterprise applications that have developed on a parallel course with relational databases. In the next section, we illustrate the information integration challenge using a scenario drawn from a real-world problem.

Scenario

To meet the needs of its high-end customers and manage high-profile accounts, a financial services company would like to develop a system to automate the process of managing, augmenting and distributing research information as quickly as possible. The company subscribes to several commercial research publications that send data in the Research Information Markup Language (RIXML), an XML vocabulary that combines investment research with a standard format to describe the report's metadata. Reports may be delivered via a variety of mechanisms, such as real time message feeds, e-mail distribution lists, Web downloads and CD ROMs.

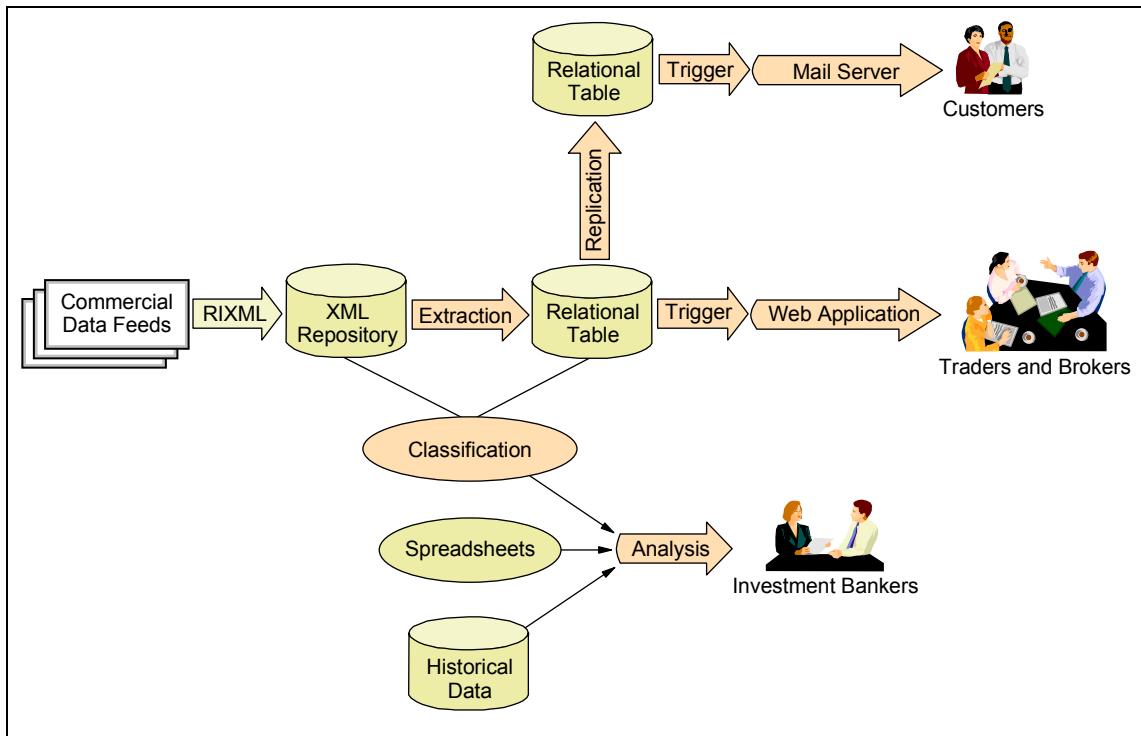


Figure A-27 Financial services scenario

Figure A-27 shows how such research information flows through the company.

- ▶ When a research report is received, it is archived in its native XML format.
- ▶ Next, important metadata such as company name, stock price, earnings estimates, etc., is extracted from the document and stored in relational tables to make it available for real time and deep analysis.
- ▶ As an example of real time analysis, the relational table updates may result in database triggers being fired to detect and recommend changes in buy/sell/hold positions, which are quickly sent off to equity and bond traders and brokers. Timeliness is of the essence to this audience and so the information is immediately replicated across multiple sites. The triggers also initiate e-mail notifications to key customers.
- ▶ As an example of deep analysis, the original document and its extracted metadata is more thoroughly analyzed, looking for such keywords as “merger,” “acquisition,” or “bankruptcy” to categorize and summarize the content. The summarized information is combined with historical information made available to the company’s market research and investment banking departments.

- ▶ These departments combine the summarized information with financial information stored in spreadsheets and other documents to perform trend forecasting, and to identify merger and acquisition opportunities.

Requirements

To build the financial services integration system on today's technology, a company must cobble together a host of management systems and applications that do not naturally coexist with each other. DBMSs, content management systems, data mining packages and workflow systems are commercially available, but the company must develop integration software in-house to integrate them. A database management system can handle the structured data, but XML repositories are just now becoming available on the market. Each time a new data source is added or the information must flow to a new target, the customer's home grown solution must be extended.

The financial services example above and others like it show that the boundaries that have traditionally existed between DBMSs, content management systems, mid-tier caches, and data warehouses are increasingly blurring, and there is a great need for a platform that provides a unified view of all of these services. We believe that a robust information integration platform must meet the following requirements:

- ▶ ***Seamless integration of structured, semi-structured, and unstructured data from multiple heterogeneous sources.*** Data sources include data storage systems such as databases, file systems, real time data feeds, and image and document repositories, as well as data that is tightly integrated with vertical applications such as SAP or Calypso. There must be strong support for standard metadata interchange, schema mapping, schema-less processing, and support for standard data interchange formats. The integration platform must support both consolidation, in which data is collected from multiple sources and stored in a central repository, and federation, in which data from multiple autonomous sources is accessed as part of a search, but is not moved into the platform itself. As shown in the financial services example, the platform must also provide transparent transformation support to enable data reuse by multiple applications.
- ▶ ***Robust support for storing, exchanging, and transforming XML data.*** For many enterprise information integration problems, a relational data model is too restrictive to be effectively used to represent semi-structured and unstructured data. It is clear that XML is capable of representing more diverse data formats than relational, and as a result it has become the lingua franca of enterprise integration. Horizontal standards such as ebXML and SOAP provide a language for independent processes to exchange data, and vertical standards such as RIXML are designed to handle data exchange for a specific industry. As a result, the technology platform must be XML-aware and optimized for XML at all levels. A native XML store is absolutely necessary,

along with efficient algorithms for XML data retrieval. Efficient search requires XML query language support such as SQLX and XQuery.

- ▶ **Built-in support for advanced search capabilities and analysis over integrated data.** The integration platform must be bilingual. Legacy OLTP and data warehouses speak SQL, yet integration applications have adopted XML. Content management systems employ specialized APIs to manage and query a diverse set of artifacts such as documents, music, images, and videos. An inverse relationship naturally exists between overall system performance and the path length between data transformation operations and the source of the data. As a result, the technology platform must provide efficient access to data regardless of whether it is locally managed or generated by external sources, and whether it is structured or unstructured. Data to be consolidated may require cleansing, transformation and extraction before it can be stored. To support applications that require deep analysis such as the investment banking department in the example above, the platform must provide integrated support for full text search, classification, clustering and summarization algorithms traditionally associated with text search and data mining.
- ▶ **Transparently embed information access in business processes.** Enterprises rely heavily on workflow systems to choreograph business processes. The financial services example above is an example of a macroflow, a multi-transaction sequence of steps that capture a business process. Each of these steps may in turn be a microflow, a sequence of steps executed within a single transaction, such as the insert of extracted data from the research report and the database trigger that fires as a result. A solid integration platform must provide a workflow framework that transparently enables interaction with multiple data sources and applications. Additionally, many business processes are inherently asynchronous. Data sources and applications come up and go down on a regular basis. Data feeds may be interrupted by a hardware or a network failures. Furthermore, end users such as busy stock traders may not want to poll for information, but instead prefer to be notified when events of interest occur. An integration platform must embed messaging, Web services and queuing technology to tolerate sporadic availability, latencies and failures in data sources and to enable application asynchrony.
- ▶ **Support for standards and multiple platforms.** It goes without saying that an integration platform must run on multiple platforms and support all relevant open standards. The set of data sources and applications generating data will not decrease, and a robust integration platform must be flexible enough to transparently incorporate new sources and applications as they appear. Integration with OLTP systems and data warehouses require strong support for traditional SQL. To be an effective platform for business integration,

emerging cross-industry standards such as SQLX and XQuery as well as standards supporting vertical applications RIXML.

- ▶ **Easy to use and maintain.** Customers today already require integration services and have pieced together in-house solutions to integrate data and applications, and these solutions are costly to develop and maintain. To be effective, a technology platform to replace these in-house solutions must reduce development and administration costs. From both an administrative and development point of view, the technology platform should be as invisible as possible. The platform should include a common data model for all data sources and a consistent programming model. Metadata management and application development tools must be provided to assist administrators, developers, and users in both constructing and exploiting information integration systems.

Architecture

Figure A-28 on page 342 illustrates our proposal for a robust information integration platform.

- ▶ The foundation of the platform is the data tier, which provides storage, retrieval, and transformation of data from base sources in different formats. We believe that it is crucial to base this foundation layer upon an enhanced full-featured federated DBMS architecture.
- ▶ A services tier built on top of the foundation draws from content management systems and enterprise integration applications to provide the infrastructure to transparently embed data access services into enterprise applications and business processes.
- ▶ The top tier provides a standards-based programming model and query language to the rich set of services and data provided by the data and services tiers.

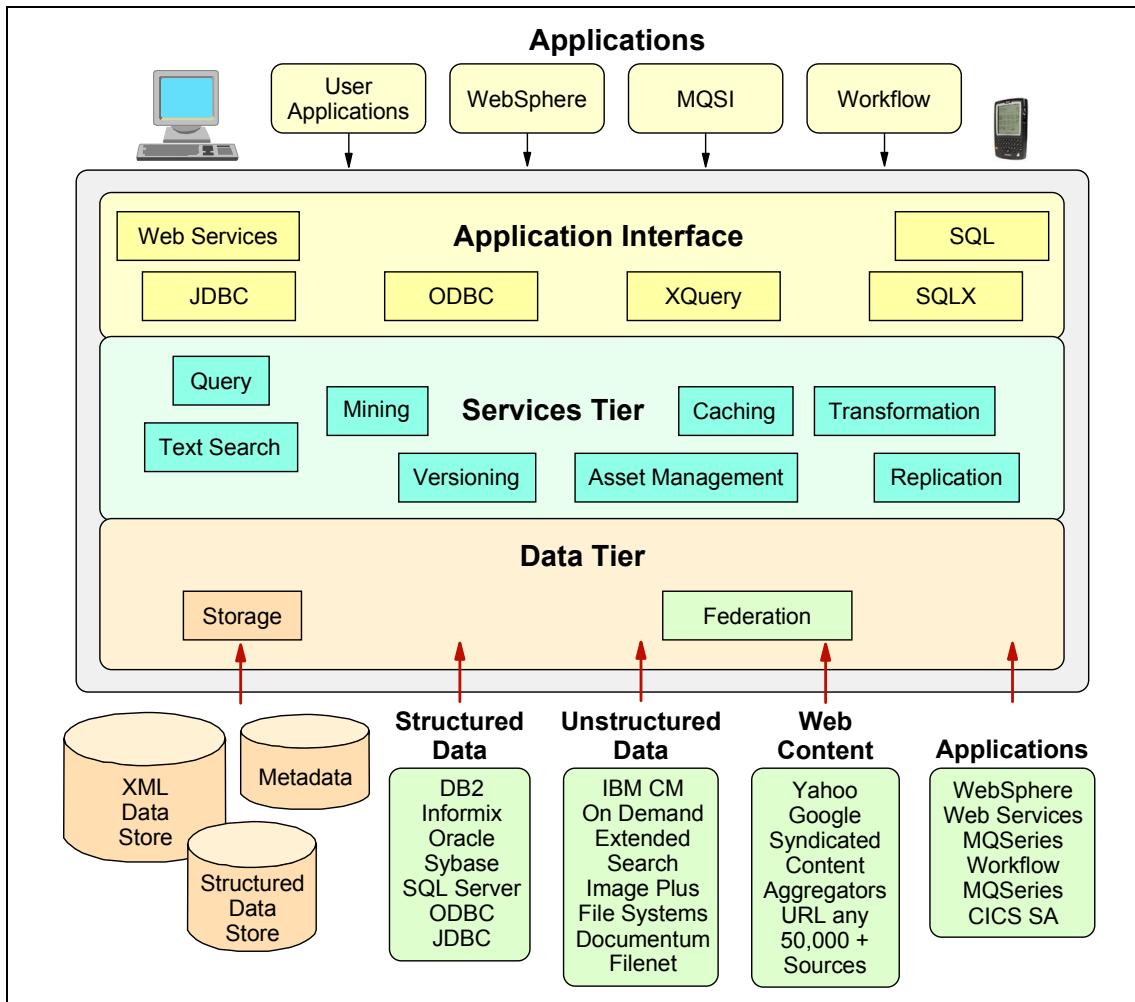


Figure A-28 An information integration platform

► The data tier

As shown in Figure A-28, the data tier is an enhanced high performance federated DBMS. We have already described the evolution of the DBMS as a robust, high-performance and extendable technology for managing structured data. We believe that a foundation based on a DBMS architecture allows us to exploit and extend these key advances to semi-structured and unstructured data.

- **Storage and retrieval.** Data may be stored as structured relational tables, semi-structured XML documents, or in unstructured formats such as byte streams, scanned documents, and so on. Because XML is the lingua

franca of enterprise applications, a first class XML repository that stores and retrieves XML documents in their native format is an integral component of the data tier. This repository is a true native XML store that understands and exploits the XML data model, not just a rehashed relational record manager, index manager, and buffer manager. It can act as a repository for XML documents as well as a staging area to merge and consolidate federate data. In this role, metadata about the XML data is as critical as the XML data itself. This hybrid XML/relational storage and retrieval infrastructure not only ensures high performance, and data durability for both data formats, but also provides the 24x7 availability and extensive administrative capabilities expected of enterprise database management systems.

- **Federation.** In addition to a locally managed XML and relational data store, the data tier exploits federated database technology with a flexible wrapper architecture to integrate external data sources. The external data sources may be traditional data servers, such as external databases, document management systems, and file systems, or they may be enterprise applications such as CICS® or SAP, or even an instance of a workflow. These sources may in turn serve up structured, semi-structured or unstructured data.

► **The services tier**

The services tier draws on features from enterprise application integration systems, content management systems and exploits the enhanced data access capabilities enabled by the data tier to provide embedded application integration services.

- **Query processing.** In addition to providing storage and retrieval services for disparate data, the data tier provides sophisticated query processing and search capabilities. The heart of the data tier is a sophisticated federated query processing engine that is as fluent with XML and object-relational queries as it is with SQL. Queries may be expressed in SQL, SQLX, or XQuery, and data may be retrieved as either structured data or XML documents. The federated query engine provides functional compensation to extend full query and analytic capabilities over data sources that do not provide such native operations, and functional extension to enable extended capabilities such as market trend analysis or biological compound similarity search.

In addition to standard query language constructs, native functions that integrate guaranteed message delivery with database triggers allow notifications to fire automatically based on database events, such as the arrival of a new nugget of information from a real time data feed.

- **Text search and mining.** Web crawling and document indexing services are crucial to navigate the sea of information and place it within a context

usable for enterprise applications. The services tier exploits the federated view of data provided by the data tier to provide combined parametric and full text search over original and consolidated XML documents and extracted metadata. Unstructured information must be analyzed and categorized to be of use to an enterprise application, and for real time decisions, the timeliness of the answer is a key component of the quality. The technology platform integrates services such as Intelligent Miner™ for Text to extract key information from a document and create summaries, categorize data based on predefined taxonomies, and cluster documents based on knowledge that the platform gleans automatically from document content. Built-in scoring capabilities such as Intelligent Miner Scoring integrated into the query language (SQLMM) turn interesting data into actionable data.

- ***Versioning and metadata management.*** As business applications increasingly adopt XML as the language for information exchange, vast numbers of XML artifacts, such as XML schema documents, DTDs, Web service description documents, etc., are being generated. These documents are authored and administered by multiple parties in multiple locations, quickly leading to a distributed administration challenge. The services tier includes a WebDav-compliant XML Registry to easily manage XML document life cycle and metadata in a distributed environment. Features of the registry include versioning, locking, and name space management.
- ***Digital asset management.*** Integrated digital rights management capabilities and privilege systems are essential for controlling access to the content provided by the data tier. To achieve these goals, the information integration platform draws on a rich set of content management features (such as that provided in IBM Content Manager) to provide integrated services to search, retrieve and rank data in multiple formats such as documents, video, audio, etc., multiple languages, and multibyte character sets, as well as to control and track access to those digital assets.
- ***Transformation, replication, and caching.*** Built-in replication and caching facilities and parallelism provide transparent data scalability as the enterprise grows. Logic to extract and transform data from one format to another can be built on top of constraints, triggers, full text search, and the object relational features of today's database engines. By leveraging these DBMS features, data transformation operations happen as close to the source of data as possible, minimizing both the movement of data and the code path length between the source and target of the data.

► The application interface

The top tier visible to business applications is the application interface, which consists of both a programming interface and a query language.

- **Programming interface.** A foundation based on a DBMS enables full support of traditional programming interfaces such as ODBC and JDBC, easing migration of legacy applications. Such traditional APIs are synchronous and not well-suited to enterprise integration, which is inherently asynchronous. Data sources come and go, multiple applications publish the same services, and complex data retrieval operations may take extended periods of time. To simplify the inherent complexities introduced by such a diverse and data-rich environment, the platform also provides an interface based on Web services (WSDL and SOAP). In addition, the platform includes asynchronous data retrieval APIs based on message queues and workflow technology to transparently schedule and manage long running data searches.
- **Query language.** As with the programming interface, the integration platform enhances standard query languages available for legacy applications with support for XML-enabled applications. XQuery is supported as the query language for applications that prefer an XML data model. SQLX is supported as the query language for applications that require a mixed data model as well as legacy OLTP-type applications. Regardless of the query language, all applications have access to the federated content enabled by the data tier. An application may issue an XQuery request to transparently join data from the native XML store, a local relational table, and retrieved from an external server. A similar query could be issued in SQLX by another (or the same) application.

Conclusion

The explosion of information made available to enterprise applications by the broad-based adoption of Internet standards and technologies has introduced a clear need for an information integration platform to help harness that information and make it available to enterprise applications. The challenges for a robust information integration platform are steep. However, the foundation to build such a platform is already on the market. DBMSs have demonstrated over the years a remarkable ability to manage and harness structured data, to scale with business growth, and to quickly adapt to new requirements. We believe that a federated DBMS enhanced with native XML capabilities and tightly coupled enterprise application services, content management services and analytics is the right technology to provide a robust end-to-end solution.

Products

We recommend that you start with the following products:

- ▶ **IBM WebSphere Studio** is an open comprehensive development environment for building, testing and deploying dynamic on demand e-business applications. Founded on open technologies and built on Eclipse, WebSphere Studio provides a flexible, portal-like integration of multi-language, multi-platform and multi-device application development tools that maximize your productivity, increase ROI, and improve overall time to value.
- ▶ **IBM WebSphere Application Server** is a high-performance and extremely scalable transaction engine for dynamic e-business applications. The Open Services Infrastructure allows companies to deploy a core operating environment that works as a reliable foundation capable of handling high volume secure transactions and Web services. WebSphere continues the evolution to a single Web services-enabled, Java 2 Enterprise Edition (J2EE) application server and development environment that addresses the essential elements needed for an on demand operating environment.
- ▶ You can use **WebSphere Business Integration Server V4.2** to quickly integrate new or existing applications or systems on diverse platforms, create and rapidly deploy new business processes, or solve a variety of business integration needs.
- ▶ **DB2 Information Integrator**—This new family of products is designed to help you integrate structured, semi-structured and unstructured information effectively and efficiently. These products, based on the previously disclosed Xperanto project, provide the foundation for a strategic information integration framework to help access, manipulate, and integrate diverse, distributed and real time data.
- ▶ **DB2 Universal Database**—DB2 Version 8.1 helps solve critical business problems by integrating information across the entire enterprise by leveraging federated Web services and XML. DB2 is delivering new federated capabilities that enable customers to integrate information as Web services. DB2 also delivers new XML enhancements that make it easier for programmers to integrate DB2 and XML information.

A.2.4 Automation

Automation leaves enterprise leaders free to focus on managing the business, rather than managing the complexities of new technology. Automation, including autonomic computing, leaves enterprise leaders free to focus on managing the business, rather than managing the complexities of new technology.

Autonomic computing

Note: The following material comes from “What you need to know now about autonomic computing, Part 1: An introduction and overview” by Daniel H. Steinberg (Director of Java Offerings, Dim Sum Thinking), published in August 2003. IBMers can find this article at <http://www-106.ibm.com/developerworks/ibm/library/i-autonom1/>.

Imagine if you could describe the business functions you want your system to provide and it just took care of itself. Needed software would be located, installed, and configured automatically. Resources would become available when they were needed and are freed when they weren't. This autonomic vision was explained by IBM's Alan Ganek in a session at developerWorks live!

Autonomic systems are able to dynamically configure and reconfigure themselves according to business needs. Such systems are always on the lookout to protect themselves from unauthorized use, to repair portions of the system that are no longer functioning, and to look for ways to optimize themselves. IBM has introduced major initiatives in autonomic computing. At this year's developerWorks Live! conference, Alan Ganek, IBM Vice President of Autonomic Computing, presented an overview that set the stage for a host of other sessions on the topic. This series presents the highlights from this year's sessions on autonomic computing at developerWorks Live!

The autonomic computing vision

Ganek's session in New Orleans came just a few days after the NCAA Final Four men's basketball playoffs concluded. He used basketball as an analogy for autonomic computing. The players think about looking for an open shot, passing to team mates, and defending against the other team. When players run the length of the court on a fast break, they concentrate on getting the basket. They don't have to think about making their heart beat faster, about altering their breathing pattern, or about altering their pupil dilation to focus on the rim of the basket. Regulating the circulation and breathing are critical to a player's success, but they should not require thought or attention. The autonomic nervous system in humans takes care of tuning these core functions and allows us to think on a higher level.

The autonomic computing vision is the analogous situation for IT. Autonomic computing allows people to focus on the big picture because the low level tasks can be “taught” to monitor and manage themselves. In a typical traditional network, there are a wide range of clients, servers, and multiple databases connected across internal and external firewalls. To manage this network you have to consider dozens of systems and applications, hundreds of components,

and thousands of tuning parameters. The goal is to have a system that can accept business rules as input to manage the increasingly complex system.

Ganek acknowledges that we've been trying to figure out how to manage complexity for years but that the current situation is different. He explains that in the mid 1990s, the Internet was deployed, but was not being used by many businesses. At that time, a big network was 10,000 ATMs connected to mainframes. "Now," he says, "you have smart cards, smart phones, laptops, and desktops, all of which come in over wireless and the Internet. You don't hit one bank of mainframes, now you might hit 10,000 servers. The Internet explosion is an explosion in complexity."

An IBM ThinkPad® sat on a desk in front of Ganek. One of the conference staff had stopped by and booted it up. Ganek pointed to it as an example of an unused asset that was just wasting cycles. He sees several advantages in implementing autonomic solutions:

- ▶ Increased ROI (return on investment) by lowering administrative costs and improving the utilization of assets. A solution should be able to locate under-utilized resources and take advantage of them in a dynamic way.
- ▶ Improved QoS (quality of service) by reducing downtime. Ganek pointed out that 40% of outages come from operator error.
- ▶ Faster time to value by accelerating implementation of new capabilities along with more accurate and immediate installation and reduced test cycles.

The components of self-management

There is not a single technology known as *autonomic computing*. Ganek points out that customers will still be buying Enterprise Resource Planning (ERP) or data management solutions. They will, however, prefer those with the characteristics of autonomic computing. Many of the sessions on autonomic computing at this conference included the image in Figure A-29.



Figure A-29 Components of self-managing systems

This diagram breaks self-management into four slices:

1. **Self-configuring components** increase responsiveness by adapting to environments as they change. Your system can add and configure new features, additional servers, and newly available software releases while the system is up and running. The key to making this process autonomic is to require minimal human involvement.
2. **Self-healing components** improve business resiliency by eliminating disruptions that are discovered, analyzed, and acted upon. The system identifies and isolates a failed component. This component is taken offline, repaired or replaced, and then the functional component is brought back online. Autonomic systems need to be designed with some level of redundancy so that this healing can occur transparently to users.
3. **Self-protecting components** anticipate and protect against intrusions and corruptions of data. This includes the managing of authentication of users for accessing resources across an array of enterprise resources. Self-protection also includes monitoring who is accessing resources and reporting and responding to unauthorized intrusions.
4. **Self-optimizing components** make the best use of available resources even though these resources and the requirements are constantly changing. Humans cannot respond quickly enough to perform these actions in a way that responds to the current system conditions. Systems must monitor and tune their storage, databases, networks, and server configurations continually.

Autonomic does not just mean automated. An automated system might simply specify that this server is assigned to a particular task between the hours of 4:00 and 7:00 and to a different task the remainder of the day. This specification may be correct and it may help to have it in place in your system. On the other hand, you may want a more business oriented rule being enforced. Your rule may be that gold level customers can expect a response to be generated within two seconds while silver customers can expect a response to be generated within six seconds.

The path to autonomic systems

Although autonomic computing is not far away, Ganek recommends a step-by-step approach to evolve your infrastructure in that direction. First you need to assess where you are in the continuum. Then you need to decide which area of complexity to tackle first. Ganek reminds the audience that “the complexity is at every level of the system. Autonomic is hardware, software, and system management.”

Ganek outlines five levels that run the gamut from manual to autonomic. He says that most organizations are at level 1. This basic level requires that the IT staff

install, monitor, maintain, and replace each system element. The second level, the managed stage, already can be implemented using many of the tools that IBM and other vendors provide. The tools help the IT staff analyze system components and use the results to decide which actions to take. Ganek says that many state-of-the-art customers are currently at this level.

Each level replaces some area of human intervention and decision making. The predictive level adds builds on the monitoring tools added in the previous level. At this third level, the system can correlate measurements and make recommendations. The IT staff looks to approve the recommendations and take actions. This leads to faster and better decision making. At level four, the staff becomes less involved in viewing the recommendations and taking actions. This is the adaptive level and features the ability of the technology to make more of the decisions automatically. Staff members spend most of their time setting the policies and managing the controls.

The *autonomic* level is level five. In many ways the technology is similar to that introduced at level four. A difference is that the IT services are now integrated with business rules. This is where you stop defining IT rules in terms of the components and tuning parameters. Now the policies are set in terms of business logic, and the IT staff focuses on tuning the system and the rules so they best support the company bottom line. As an example, if a Web site supports free content and subscriber-only content, then a rule might specify that resources should be allocated so that the user experience for subscribers is at a certain level even if that means degrading the experience for non-paying site visitors.

The near future for autonomic computing

In order to deploy autonomic solutions, Ganek explains, there are core capabilities that must be provided. Although this list will change, he suggests an initial set that includes solution installation, common system administration, problem determination, monitoring, complex analysis, policy based management, and heterogeneous workload management. For each of these areas, IBM is looking at a technology to advance those capabilities.

Ganek cited some of the existing IBM projects. There is a partnership with InstallShield, designed to make installation more regular and more familiar. This process will help users understand what is to be installed and what must be installed along with a recognition of dependencies. IBM is participating in the Java Community Process JSR-168. By using Portal technology, the administrative console will become consistent with the same terminology and will interface across applications. Using the agent standards described in JSR-87, IBM has created ABLE, an Agent Based Learning Environment, and will use intelligent agents to capture and share individual and organizational knowledge.

These agents will be used to construct the calculation tools needed for autonomic computing.

You can track many of the latest autonomic releases on the newly created autonomic zone on alphaWorks. Other articles in this series describe the grid structure that underlies much of the dynamic sharing of resources and the autonomic cycle that will work to monitor each autonomic element, analyze it, plan for change, and execute the change. This will be accompanied by concrete examples for logging and tracing and for workload management.

Infrastructure—A look at the control loop, the toolkit, and policies

At the heart of self-managing systems are the policies used to manage them. Policies have to enable the system to choose among potentially contradictory guidelines by deciding which option would best help achieve business objectives.

As you make the transition from manual to autonomic, you trust your system to make increasingly complicated recommendations and to take action upon them. In this article, you'll read about the cycle used to manage elements in an autonomic system and about the policies used to make decisions about what action to take. This article is based on two presentations at this year's developerWorks Live! conference. David W. Levine presented an overview of the process and the tools you will be able to find on the alphaWorks site this summer, in his session titled "A toolkit for autonomic computing." David Kaminsky presented the four components of a policy and how they are used in autonomic computing in his session, "Policy-driven Computing—The Brains of an Autonomic System."

The autonomic cycle

Autonomic elements have two management tasks: they manage themselves and they manage their relationships with other elements through negotiated agreements. An autonomic element contains a continuous control loop that monitors activities and takes actions to adjust the system to meet business objectives. Autonomic computing components are available as part of the Emerging Technologies Toolkit (ETTK) on alphaWorks. You can use the ETTK to experiment with components that serve as the building blocks for self-management: monitoring, analysis, planning, and execution. The architecture is summarized in Figure A-30 on page 352.

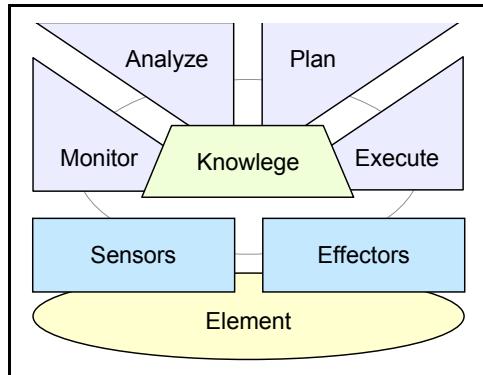


Figure A-30 The autonomic cycle

At the bottom of the diagram is the element that is being managed. It is linked to the control loop with sensors and effectors. You can think of these as high-level getters and setters. The sensors are used to provide information on the current state of the element and the effectors are used to modify this state in some way. The sensors and effectors are exposed to the control loop as Web services.

Refer to the control loop by the acronym *MAPE*: *monitor*, *analyze*, *plan*, and *execute*. In the middle of the diagram is knowledge. Knowledge is data with context and structure. The number 1.7 is a piece of data, but it isn't knowledge. The fact that we rejected an average of 1.7 requests per minute in the last hour is knowledge. Having a context and structure for the data allows users to write components which share data. This is crucial because over time, a complex system won't be solved by one programmer. Just as important is the ability to separate the people who write the tools that provide information from the people who write the policies. The collection of knowledge helps with this separation.

Monitoring, the fetching of low-level information, is where most of the knowledge comes from. This is not simply the persistence of every piece of data that can be gathered. You want to be careful not to overload the system with too many pieces of uninteresting data. You might decide to take readings every 10 seconds and persist the data two minutes at a time into a temporary store. When something goes wrong, you might add the relevant readings to the knowledge base. The analysis tools distributed with the Autonomic Computing Toolkit include prediction algorithms, modeling tools, and a math library. More specifically, these include workload forecasting, stochastic modeling, and optimizers.

Once you know what needs to be done, what remains is to make plans and to make changes. These steps are based on rules engine technology. You can evaluate policy against the current operating conditions that you have measured

and analyzed in the previous steps. You can separate out how you react to events and new agreements by setting up policies. The goal with rules is to capture interesting behavior without getting caught in overly long-running inferences. The infrastructure for planning and executing is consistent with OGSA and with the W3C policy initiative discussed in the next section.

Policy-driven computing

From the autonomic computing standpoint, policies can be viewed as a set of considerations designed to guide decisions on courses of actions. It is a piece to guide the decisions that guide how a system behaves. Policy-driven computing is the brains of an autonomic system. This does not imply that there are no conflicts among policies that apply to a given situation. The policies are considerations guiding the system, requiring that an interpreter resolves any conflicts.

One of the advantages is that when you can set policies and rules for prioritizing them, you reduce the need to configure individual resources. You are also allowing the system to make adjustments to the configuration when the load on the system changes. Practically speaking, your goal is to take a service-level agreement and abstract the services you are going to provide as service-level objectives. These objectives are then mapped to policies. You can then monitor against the objectives to see whether they are being met. One example is a policy that promises gold clients one second response time, silver clients three seconds, and bronze clients five seconds. You can monitor the response time for each customer level, and dynamically reallocate servers if you are meeting the bronze objectives but not the gold objectives.

Before proceeding, you need to consider whether you are using the same notion of policy as the autonomic computing team. As the speaker points out, there are many different ideas of what is meant by policy. The IETF definition is characterized as guiding actions. The British-based Ponder looks at a policy as a rule that defines a choice in how the system behaves. The WS Policy group looks at a policy statement as being made up of the more atomic policy assertions. These represent preferences and can be thought of as properties that can be set. For IBM, policies are used to guide decisions and should encapsulate some business requirement.

The components of a policy

IBM views a policy as a “four-tuple” made up of scope, preconditions, measurable intent, and business value. In the autonomic computing cycle, the policy sits inside of the planning phase. It configures the monitoring and execution phases to meet the objectives of the policy. In subsequent cycles, the analysis phase is where it is determined if the policy has been met.

Scope and preconditions are the first components of the policy “four-tuple.” With scope, you identify what is and is not subject to the intent of the policy. This

includes which resources are needed and perhaps which policies are applicable. For example, if you have a backup policy, you should define the scope so that it is clear whether it is applied to a database or to a Web server. A policy may be capability-based. In this instance, your backup policy might be applied to any resource capable of performing a backup. Preconditions help the system decide which policies are relevant. Perhaps you want to employ one backup strategy while the system is heavily utilized and a different backup strategy when it is not. You might determine which strategy to use based on the state of a particular system. For example, if an average of more than two requests each minute are received, then employ the first strategy. Your choice might also depend on something as simple as the time of day. Perhaps you choose one strategy during business hours and a different one overnight. The preconditions help you specify the situations in which a particular policy is to be applied.

The remaining components of the policy “four-tuple” are used to define what is the intended result of the policy and to provide further guidance in selecting among apparently conflicting policies. Measurable intent is where you specify what you are trying to accomplish. Examples include “perform a backup” or “provide a one second response time.” Business value is used to optimize resource use. You may have one policy that describes a backup and another that prescribes response time. Following one policy may mean that you have to violate another. Business value helps the system select among these applicable and relevant policies.

Help is on the way

In order to build and support policies, you need tool support. These tools help you define a policy and store it while it is being validated. You will then need to use either push or pull to enforce the policy. In defining a policy, the key issue is ensuring the policy is relevant to the resources on which you are setting the policy. One strategy is for the resource to tell the tool what sort of policies it can handle. There is a need, when setting policy, to understand the instrumentation of the state of the system. In addition, you want to provide transactional coherence. It is possible that a collection of policies either don't make sense or could be destructive if they are not all deployed together. Transactional coherence enforces these requirements.

Research on policies also includes validation, distribution, and security. The validation step involves conflict resolution. Conflicts could be static or dynamic. You may ask whether an individual can set a particular policy, or you can ask whether an individual can set a particular policy at the current state of the system. Policies are an avenue for attacking a system. It is important to include security and authentication when planning for distribution of a system.

Between the second half of this year and the middle of next year, IBM will be working on Policy editors and validators. The goal is to package many of these

technologies with the ETTK to simplify the development of policy-enabled systems. The releases will include rule evaluation engines, business policy to IT policy translations, SLA compliance monitors, and WS Policy enablement. The IBM policy efforts are driven by standards, and the engineers are working on tools that support and facilitate the implementation of those standards. In future articles in this series, we'll look at other standards-based efforts in autonomic computing. These include work on the grid infrastructure for autonomic computing and on deployment of autonomic solutions.

The underlying grid—Grid computing, standards, workload management, and the future of WebSphere Application Server

Autonomic systems are self-healing and self-optimizing. Sometimes they need to use resources that are distributed across a network. Grid computing provides the facility for optimal use of resources in a robust and flexible system.

Grid computing lets virtual collaborative organizations share applications and data in an open, heterogeneous environment. Autonomic computing features self-managing systems that benefit from the grid support for sharing, managing, and providing access to resources. This linked management can be used to increase the quality of service and to optimize the available resources. Providing access across the grid supports on demand computing and utility models. A grid allows you to increase your capacity by exploiting distributed resources to provide capacity for high-demand applications. Distributed resources also provide reliability and availability and reduce time to results through computations that are executed more frequently or on customer demand.

Grid computing is being integrated into the design of IBM WebSphere Application Server Version 5. Version 6 will be compliant with Open Grid Services Architecture (OGSA). You will be able to build grids using Application Server, and you will be able to add Application Server to existing grids. In addition, DB2, the IBM eServer product line, and IBM TotalStorage® solutions will be OGSA compliant and will also be easily integrated into grids. This article on grid computing is based on two presentations from this year's developerWorks Live! conference in New Orleans. Marlon Machado and Christopher Walden discussed the changes to Application Server to take advantage of grid computing in their presentation, "Grid Computing in the Real World: WebSphere Application Server V4.0's new system management architecture." Matthew Haynos looked at the grid computing standards in his talk, "Working with Open Grid Services Architecture (OGSA)."

Application Server version 5 administration

In rethinking Application Server, the architects decided to adopt a grid-like approach to achieve performance, capacity, and failover support. In particular, there is a move away from a single administrative server and a centralized

repository for configuration, clustering, and work load management. In the Application Server Version 4 model, there was a single administrative domain, and all nodes were registered in a relational database. This strategy was robust, but there was a single point of failure, and customization had to be done by hand.

In Application Server Version 5, systems administration is an extension of the application. You still have nodes, but they aren't necessarily clones. The administration is divided into cells and nodes. Nodes are managed servers on a physical box that control processes. A cell is a logical node container. A set of cell managers administers one or more nodes. The configuration data is stored within each process. Each process is self-sufficient in managing its own resources. The configuration data is kept on a set of XML files. Cell managers are synchronized using JMX and distribute the administration of the system. They make sure configuration and binaries are updated using asynchronous communication.

A set of design principles guided the redesign of Application Server:

1. **Processes are unreliable**—The product has to be able to function even if a managed process isn't executing as expected.
2. **Communication channels are unreliable**—The goal is for the system to continue to function if a component fails.
3. **Configuration should be separated from operations**—Operations are supported by JMX and will be dynamic and synchronous if possible.
4. **Configuration is document-based**—This led to the decision to store configuration values in XML files.
5. **Isolate administrative functions**—Application servers and applications can be somewhat independent if they are isolated from each other.
6. **Make backups before making changes**—This is always good advice for human administrators, and it now applies to application servers and administrative functions. Hand-in-hand with this is the extensive use of logging throughout the system.

Other changes to Application Server result from the move toward a grid architecture. Cell managers and node managers work together to distribute the application. Processes are self sufficient but loosely coupled to nodes. All updates are automatically distributed across the domain. The message-driven, grid-like architecture is peer-to-peer, based on the JXTA V1.0 protocols spec. The message binding and file transfer layers are used to publish and synchronize configuration data, to launch administrative processes, and to support services such as name, security, and location. Concurrent performance, capacity, and failover support are addressed through graph-traversing algorithms and combinatorial optimization. Think of your processing in terms of processes and components and not just on a box they are running on. Your JMS and

application server processes can share nodes, or they can be separated if needed.

Workload management

A concrete example of the advantages of grid computing is provided by the rethinking of workload management (WLM) and scalability. Now that there is no single Application Server repository, configuration data is shared among nodes. This means that WLM must now be asynchronous and adaptive. This also requires that WLM be factored into your decisions of how you set up your system. Load balancing becomes the foundation for partitioning and grid-like behavior. Requests are dynamically distributed among clones according to availability.

IBM identified four requirements for WLM:

1. **Integration**—You need to manage HTTP, IIOP, JMS, or JavaMail requests as they move through the enterprise.
2. **Load balancing**—Balance the workload based on the resources available at any given time.
3. **Failure identification**—Describe and document what went wrong and mention all of the resources involved.
4. **Standalone availability**—WLM functionality should exist even in absence of code.

The WLM controller defines controllable operation goals. It takes the input for the attributes of the defined goals and uses a dynamic routing table to communicate to the elements. Here's a quick look at the WLM algorithm.

Initially you decide which elements and clones are heavier for this weighted round-robin algorithm. All weights are collected in a weight set for the cluster. The weight of a clone is decreased by one for each processed request until it equals zero. At this point, no more requests are sent to that clone. The process goes on until the entire weight set equals zero. The WLM routing algorithm can be overridden with in-process optimization. If a request from a client is sent to an object that is already in the same process, the weights are not decreased. A *prefer-local* optimization helps ensure that all requests are sent to the same node if possible. You can also override with transactional affinity. In a transaction, all requests will be processed on the same clone if possible. The weights are decreased, but affinity will override zero or negative weights. In other words, the core algorithm can be adapted in situations in which another algorithm is preferable.

The notion of partitioning includes and extends what we used to call scaling. Now you can organize topologies according to business needs and not just by performance and capacity requests. Think of partitioning in two directions. You

have different layers, including a Web server, servlet engine, EJB engine, and a database. You also have columns that include one component from each layer. You might have a column that includes one Web server, one database, and so on. Elements from a single column may be located on different machines. If you are familiar with Application Server Version 4, then vertical partitioning is analogous to what you thought of as horizontal scaling. Horizontal partitioning has layers that are independent from each other that are cloned and very granular. The key to performance in Application Server Version 5 is a flexible topology.

Grid computing standards

OGSA provides a services-oriented virtual organization. Providing virtualized services across an enterprise provides a common base for autonomic management solutions. IBM will OGSA-enable servers, storage, and networks this year. On top of these will sit OGSA-enabled versions of security, workflow, database file systems, directory services, and messaging. You can imagine a Web services layer that sits on top of these functional elements and is used to communicate with the OGSA layer.

The Open Grid Services Infrastructure (OGSI) can be thought of as Web services++. It improves on some of the pieces of Web services, including discovery, lifecycle, registry management, factory, notification, and HandleMap. Grid services are transient, and many of these extra pieces help Web services interact with transient services. The other additions to Web services provide data and state qualities. For example, HandleMap helps you get a pointer to a Web service. You will need lifetime management interfaces to allow the service to create and destroy itself.

Recall the monitor, analyze, plan, and execute cycle for autonomic elements described in the previous article in this series. There you used Web services to communicate with the autonomic element. The sensors and effectors reported back from the element to the managing cycle, and the effectors changed the state of the element based on the plan being executed. The grid services allow you to manage and optimize elements that you bring in and out of your system from across a network. In the final article in this series, you will see concrete applications for deploying applications and for logging services.

Deploying and logging—A support structure for self-healing applications

If an application is to be self-healing, it must be able to dynamically deploy updates and other instances. It must also be able to locate problems, recommend actions, and execute them. The final article in this series on autonomic computing looks at deploying and logging as essential support technologies.

Previously we looked at the theory and guiding ideas of autonomic computing and the underlying infrastructure. To have self-managing systems, you will need to be able to identify and diagnose problems and to deploy software designed to remedy these problems. This article talks about challenges in deploying applications and consistent logging strategies based on two talks in the autonomic computing thread from this year's developerWorks Live! conference. Heng Chu addressed the challenges of deployment in his session, "Common Software Deployment: An Enablement for Autonomic Computing." Dave Ogle outlined the benefits and difficulties in settling on a common logging format in his session, "Unified Logging and Tracing—A Building Block for Autonomic Computing."

Software deployment

Deploying software isn't just a matter of creating a CD and copying it on target machines. Here is a six-stage process for software deployment.

1. **Create the software packages**—Packaging should be done throughout the life cycle. You might be creating a product CD, or clients might want to repackage your software within their install base before they deploy it.
2. **Analyze the environment**—In a complex IT environment, you need to check dependencies on hardware, the operating system, or previously installed software. You might also need to determine if migration is needed.
3. **Plan for deployment**—Depending on your analysis, you might need to determine a migration path. Also, you need to identify where your software components will be installed.
4. **Install the packages**—This step includes the processes of moving software onto or off of a particular machine. This step might be where you install, uninstall, migrate, repair, rollback, or commit various components.
5. **Configure the software**—You ensure that the software was properly installed. You might need to configure a product to work properly in an environment or with other components.
6. **Verify the deployment so the software is ready to use**—You might smoke test the installation, verify the package is intact, or check that the entire suite has been installed and configured to handle end-to-end transactions.

Today, most installation is at the basic/managed end of the manual-to-autonomic spectrum. At the *basic level*, a highly skilled IT staff reads through the documentation and performs the installation. At the managed level, the IT staff analyzes needs and uses install tools to automate the installation step of the deployment process. At the predictive level, the IT staff allows the system to recommend deployment actions. The staff approves the recommendations and uses the installation tools to perform the install. At the adaptive level, the IT staff manages performance against service level agreements and allows the system

to understand, correlate, and take deployment actions while taking dependencies into account. Finally, at the autonomic level, the system dynamically deploys components based on business rules expressed as policies.

To automate this dynamic deployment of software, IBM is defining the concept of an installable unit and is treating a hosting environment as an autonomic computing-managed element, as shown below.

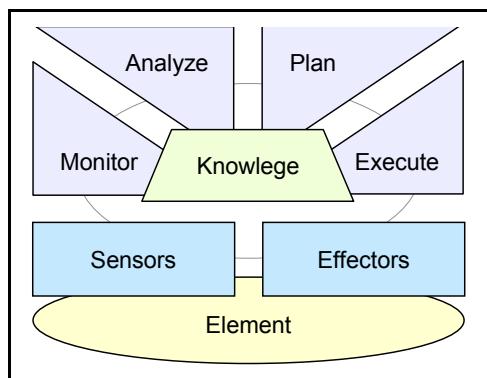


Figure A-31 The autonomic cycle

The *monitor phase* is used to gather inventory of existing software and configuration. If you know the inventory has changed, the analyze phase can be used for dependency checking, to verify the integrity of the environment, and to see that introducing new software will not destabilize the environment. The planning phase includes target specification, the choice of migration paths, and establishing the configuration for each policy. The execute phase is where you initiate installation, configuration, and verification. The sensors and effectors are the link between this managing cycle and the element being managed. Here, the sensors advertise installed components and their configuration, while the effectors are a Web service interface between the execute phase and the element that actually carries out these tasks. You can think of the knowledge that sits in the center of this scenario as the repository for information about installed software and its configuration and dependencies.

Unified logging and tracing

When something goes wrong with a complex system, locating and identifying the problem can be a nightmare. Your Web application might use a Web server, a database, storage, and other components. Maybe you are supporting more than one database system or more than one type of server. Each product uses its own log file format, and each defines and uses its own events. A failure on one part of the system might actually be the result of a failure somewhere up the message

chain. Lack of a common format and vocabulary makes it difficult to write programs to debug or tune your running application.

One solution is to create a common format for reporting errors. For the most part, most messages are reporting the following three pieces of information:

- ▶ **Observing component**—The ID of the component that is seeing a problem.
- ▶ **Impacted component**—Which component is having the problem.
- ▶ **Situation details**—An explanation using common terms of what occurred.

Being able to uniquely identify a component is critical. You want to be able to correlate between reports that originate with two different components. You want to be able to determine whether the impacted and observing components are actually the same. Also, if you are going to automate the process where action is taken, you need to be able to identify the component that needs to be acted upon.

The last leg of the trio requires a consistent way to report common situations. There tends to be creative authorship with variations, even within a single product, on how problems are reported. Across products this becomes even more difficult. IBM looked at thousands of log files to establish a small set of canonical situations. Surprisingly, the result was less than two dozen categories of logged events. Within each category there were many different ways of saying the same thing. Few messages could not be categorized. The situation taxonomy and grammar includes the situation category, the disposition, the task, and the reasoning domain. The initial set of situations includes start, stop, feature, dependency, request, configure, connect, and create. Take a look at that list and consider how many of the situations you encounter fit into one of those categories. Now think about the number of different words you have used to describe any one of them.

The challenge is how to achieve common situations and data. Customers have servers, databases, and other pieces from a variety of vendors. These other vendors might not agree to the common format, and many of the customers might not need to update their systems to include compliant components. For now the solution is to install an adapter that sits on your side of the log file. The adapter translates the current log output into the common situation format. You can get the log information from any element in the common format. This lets you use the autonomic computing cycle to manage the component. An analysis engine can work on the log files. The knowledge base can consist of a database of symptoms. This symptom database will benefit from a common format of information it delivers. To keep this explosion of data from overloading the system, we want to do as much analysis as possible close to the source and filter the data to make it more manageable.

How Tivoli software products support the IBM Autonomic Computing Initiative

Note: The following is from the IBM whitepaper of the same name, published in October 2002. IBMers can find this at [ftp://ftp.software.ibm.com/software/tivoli/whitepapers/wp-autonomic.pdf](http://ftp.software.ibm.com/software/tivoli/whitepapers/wp-autonomic.pdf).

The high tech industry has spent decades creating systems of ever-increasing complexity to solve a wide variety of business problems. Today complexity itself has become part of the problem. After deployment, hardware and software problems occur, people make mistakes and networks grow and change. Improvements and changes in performance and capacity of IT components can require constant human intervention. A machine waiting for a human to tune it and fix it can translate into lost dollars.

With the expense challenges that many companies face, IT managers want to improve the return on investment of IT by reducing the total cost of ownership, improving the quality of service and managing IT complexity. Autonomic computing helps address these issues and more by using technology to manage technology. Autonomics is a term derived from human biology. In the same way that your body's autonomic system monitors your heartbeat, checks your bloodsugar level and keeps your body temperature at 98.6° Farenheit without any conscious effort on your part, autonomic computing components anticipate needs and resolve problems—without human intervention.

IBM products with autonomic capabilities can deliver customer value with their predictive and proactive functions that anticipate changing conditions and problems. This paper defines the customer value of autonomic computing, the requirements for achieving an autonomic environment, the steps for successful implementation and the products that are making this computing concept a reality.

The customer value of autonomic computing

Autonomic computing was conceived of as a way to help reduce the cost and complexity of owning and operating the IT infrastructure. In an autonomic environment, IT infrastructure components—from desktop computers to mainframes—are self-configuring, self-healing, self-optimizing and self-protecting. These attributes are the core values of autonomic computing.



Figure A-32 Components of self-managing systems

- ▶ **Self-configuring**—With the ability to dynamically configure itself on the fly, an IT infrastructure can adapt—with minimal human intervention—to the deployment of new components or changes in the IT environment.
- ▶ **Self-healing**—A self-healing IT infrastructure can detect when IT components fail and can cure or work around those component failures to provide continued availability of business applications.
- ▶ **Self-optimizing**—Self-optimization is the ability of the IT environment to efficiently address resource allocation and utilization with minimal human intervention.
- ▶ **Self-protecting**—A self-protecting IT environment can detect hostile or intrusive behavior as it occurs and take autonomous actions to make itself less vulnerable to unauthorized access and use, viruses, denial-of-service attacks and general failures.

In an autonomic environment, components work together and communicate with each other and with high-level management tools. They regulate themselves and, sometimes, each other. They can proactively manage the network while hiding the inherent complexity of these activities from end users.

The IBM view of autonomic computing is to make its software behave automatically and bring the autonomic systems management capability to the infrastructure, enabling the IT environment—including systems management software—to configure, optimize, heal and protect itself.

Typically a complex IT infrastructure is managed using a set of IT management processes. Industry initiatives, including IT Infrastructure Library (ITIL) and IBM IT Process Model, define best practices for managing the IT environment.

Figure A-33 on page 364 shows an example of a typical process flow for incident management, problem management and change management. The actual

mechanics of how these flows are implemented in a particular IT organization can vary, but the basic functionality is usually the same.

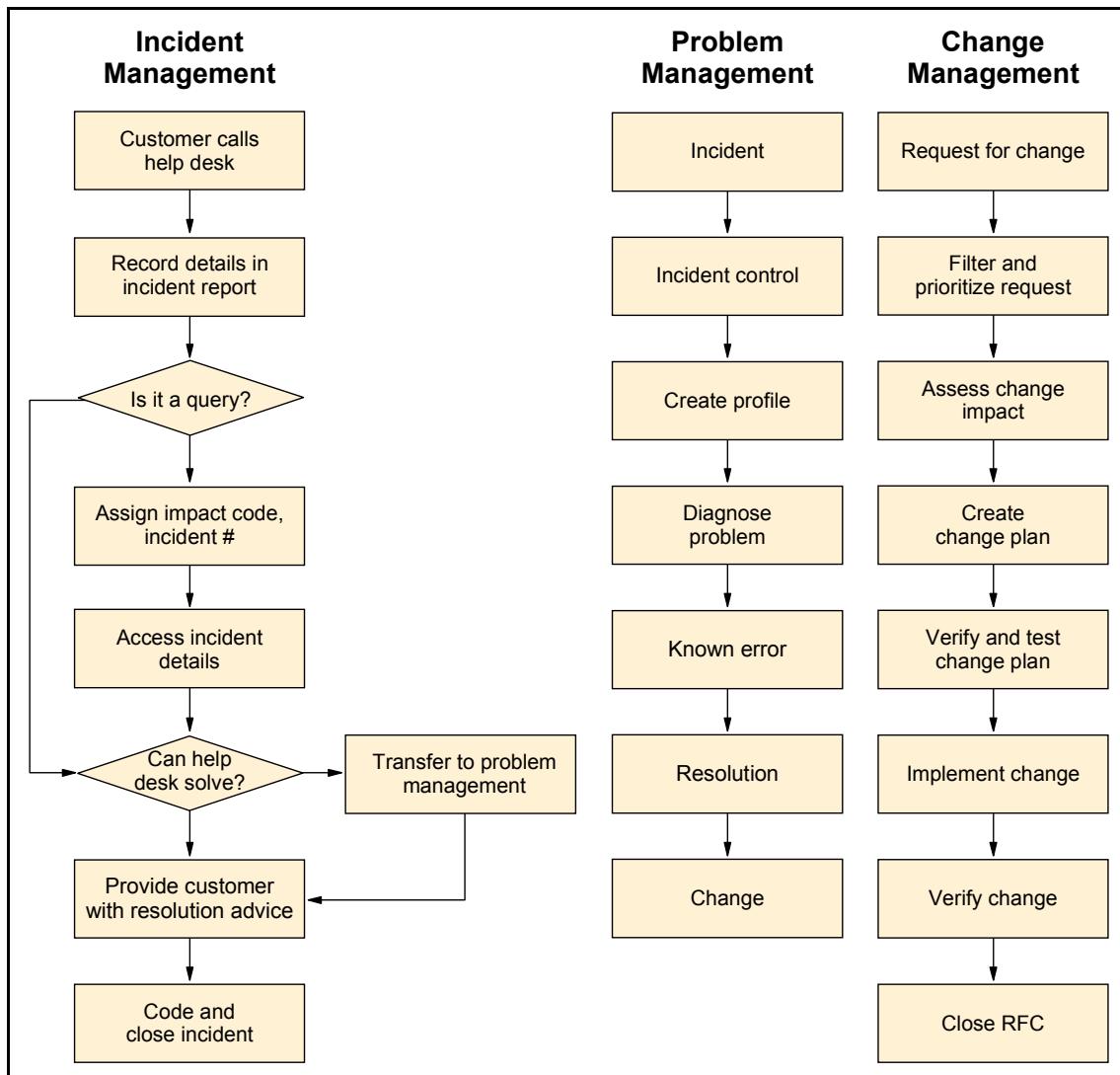


Figure A-33 A typical process flow for incident management, problem management, and change management

The efficiency and effectiveness of these processes are typically measured using metrics like elapsed time of a process, percentage executed correctly, skill requirements, average cost of execution and so on. Autonomic computing

technology can help improve the efficiency and speed with which these processes can be implemented by automating some steps in the process.

- ▶ **Quick process initiation**—Typical implementations of these processes require a human to initiate the process (create the request for change, collect incident details, open a problem record). This usually requires the IT professional to spend time gathering the right information. In a self-managing system, components can initiate the processes based on information derived directly from the system. This helps reduce the manual labor and time required to respond to critical events.
- ▶ **Reduced time and skill requirements**—Tasks or activities in these processes usually stand out as skills-intensive, long-lasting, and difficult to complete correctly the first time because of system complexity. In a change management process such an activity is change impact analysis, and in problem management such an activity is problem diagnosis. In self-managing systems, resources are instrumented so that the expertise required to perform these tasks can be encoded into the system, helping reduce the amount of time and skills needed to perform these tedious tasks.

The self-managing capability of the IT environment helps improve responsiveness, reduce total cost of ownership, and improve time to value. It can help reduce the total cost of ownership because the IT professional can complete the IT processes at a low average cost, and it can help accelerate time to value because it reduces the time it takes to execute an IT process.

The remainder of this section discusses the autonomic computing technology and tools that help make it possible.

Autonomic computing architecture concepts

The architecture shown in Figure A-34 identifies the required architectural elements in an autonomic environment. The architecture is organized into two major elements—a managed element and an autonomic manager.

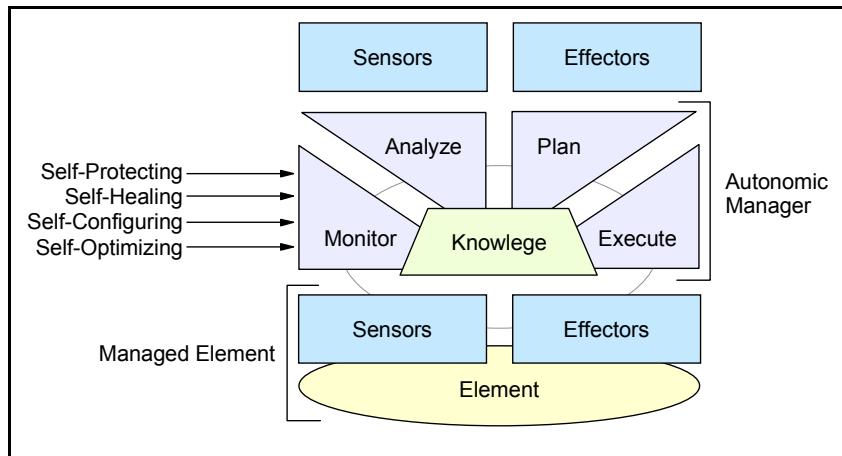


Figure A-34 Structure of self-management technologies

The managed element is the resource being managed. At this level of the architecture, the element targeted by management could be a single resource or a collection of resources. The management element exports sensors and effectors. Sensors provide mechanisms to collect information about the state and state transition of an element. Effectors are mechanisms that change the state of an element.

Sensors and effectors represent the instrumentation interface that is available to an autonomic manager. The autonomic manager is a component that implements the control loop. The architecture decomposes the loop into four parts:

- ▶ **Monitor**—Mechanisms that collect, aggregate, filter, manage and report details (metrics, topologies and so on) collected from an element.
- ▶ **Analyze**—Mechanisms to correlate and model complex situations (time series forecasting, queuing models). These mechanisms allow the autonomic manager to learn about the IT environment and help predict future situations.
- ▶ **Plan**—Mechanisms to structure the action needed to achieve goals and objectives. The planning mechanism uses policy information to guide its work.
- ▶ **Execute**—Mechanisms that control the execution of a plan with considerations for on-the-fly updates.

The monitor, analyze, plan and execute parts of the autonomic manager relate to the functionality of most IT processes. For example, the mechanics and details of IT processes like change management and problem management are different, but it is possible to abstract these into four common functions—collect the details, analyze the details, create a plan of action and execute the plan. These

four functions correspond to the monitor, analyze, plan and execute components of the architecture.

The analyze and plan mechanisms are the essence of an autonomic computing system, because they encode the know-how to help reduce the skill and time requirements of the IT professional.

The knowledge part of the autonomic manager is where data and information used by the four components of the autonomic manager is stored and shared. Knowledge that can be found here includes policy, topology information, system logs and performance metrics.

The architecture prescribes a second set of sensors and effectors. This second set enables collaboration between autonomic managers. Autonomic managers can communicate with each other in a peer-to-peer context and with high-level managers.

Each autonomic self-management attribute of self-configuring, self-healing, self-optimizing, and self-protecting is the implementation of the intelligent control loop (in an autonomic manager) for different operational aspects of configuration, healing, optimization and protection. For example, an autonomic manager can self-configure the system with the correct software if software is missing. By observing a failed element, it can self-heal the system by restarting it. It can self-optimize the current workload if increased capacity is observed. If an intrusion attempt is detected, it can self-protect the systems by blocking the intrusion at the perimeter and by verifying the resource.

Autonomic computing in the IT environment

To understand how autonomic computing plays a role in different parts of the IT environment, it is important to view the IT environment at different levels.

Self-management within each level involves implementing control loops to allow individual resources, composite resources and business solutions to monitor, analyze, plan and execute changes to their environment.

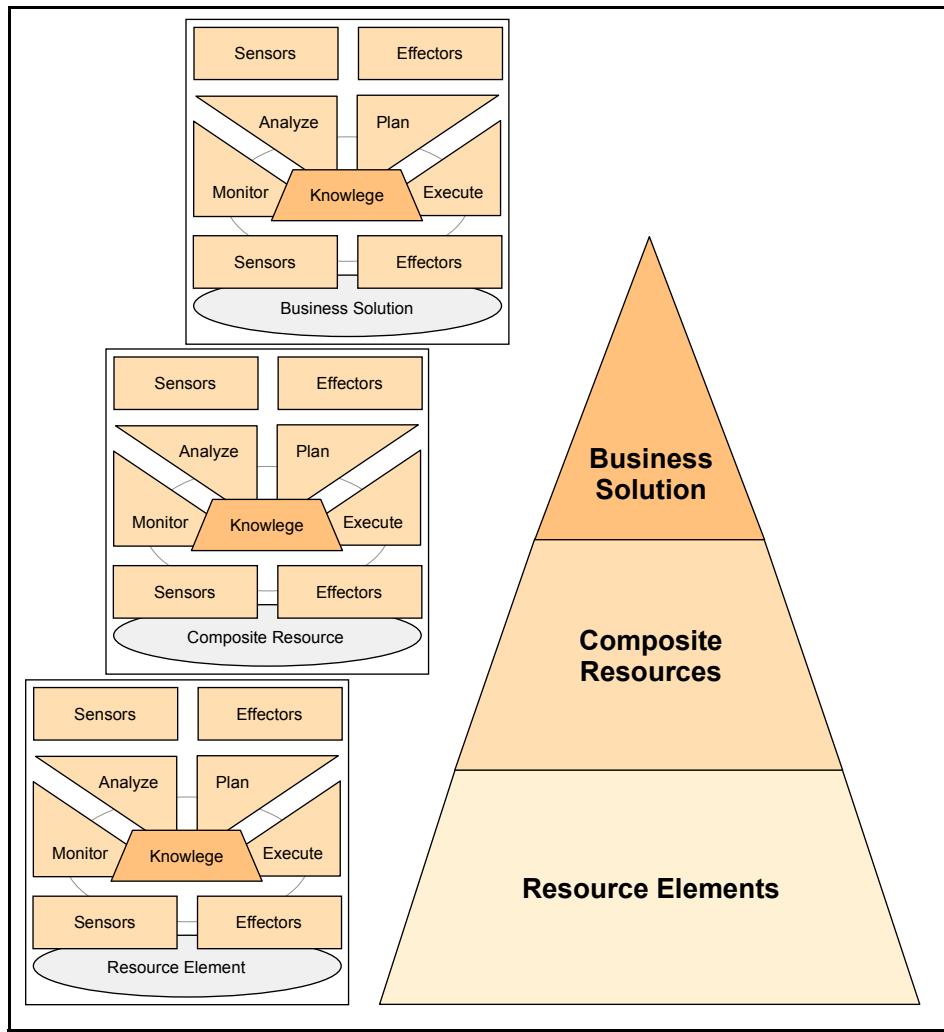


Figure A-35 *Self management within each level of the IT environment*

IBM provides a suite of management products that helps enable automation of routine management tasks for individual resource elements. IBM products, including the IBM Tivoli Monitoring family, IBM Tivoli Configuration Manager, IBM Tivoli Access Manager, and IBM Tivoli Storage Manager, begin to bring self-managing capabilities to the IT infrastructure for resource elements (systems, applications, middleware, networks and storage devices). IBM is working through IBM Server Group, IBM Software Group, and a variety of third parties to embed the appropriate technologies and enable resource elements to participate in the autonomic IT infrastructure.

At the composite resource level, the evolution to autonomic computing is enabled by the evolution to transaction-based management. In the past, resource elements were traditionally grouped by type (all servers), by location (all servers within a department or facility) or by function (all Web servers). As enterprises develop e-business environments, resources are increasingly aggregated within a transactional context spanning heterogeneous resources. For example, servers, applications, databases and storage devices that touch e-business transactions would be grouped separately from those assigned to human resources. If the composite resource grouping is homogenous (such as a server cluster) or heterogeneous (such as a Web server, database and storage system), the performance and availability requirements of different transaction types drive the autonomic activity on individual resource elements. The attainment of service-level objectives for IT transactions causes resources to be dynamically assigned, configured, optimized, and protected for changing business workloads. IBM Tivoli Monitoring for Transaction Performance, IBM Tivoli Storage Resource Manager, IBM Tivoli Identity Director, and Tivoli Configuration Manager are examples of IBM products that work together to enable the evolution to autonomics at the composite resource layer.

The highest layer of the IT environment is a business solution, such as a customer care system or an electronic auction system. The business solution layer requires autonomic systems management solutions that comprehend the state of business processes—based on policies, schedules, trends and service level objectives and their consequences—and drive the appropriate behavior for transactional systems and their underlying individual resources. Business-aware IBM products include IBM Tivoli Service Level Advisor, IBM Tivoli Business Systems Manager, and IBM Tivoli Systems Automation for S/390®.

Autonomic computing levels

Making the IT infrastructure autonomic is an evolutionary process enabled by technology, but it is ultimately implemented by each enterprise through the adoption of these technologies and supporting processes. Figure A-36 on page 370 illustrates how an IT environment evolves towards a truly autonomic environment, from basic through managed, predictive, adaptive, and finally to a fully autonomic e-business environment.

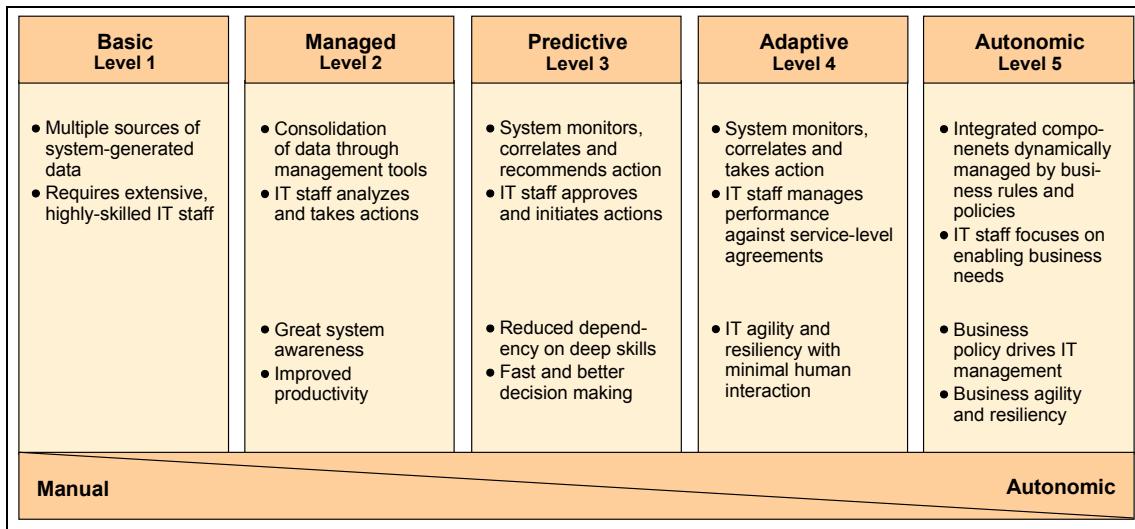


Figure A-36 How an IT environment evolves towards a truly autonomic environment

1. The basic level represents a starting point where some IT environments are today. Each infrastructure element is managed independently by IT professionals who set it up, monitor it and eventually replace it.
2. At the managed level systems management technologies can be used to collect information from disparate systems onto fewer consoles, helping reduce the time it takes for the administrator to collect and synthesize information as the IT environment becomes more complex.
3. At the predictive level new technologies are introduced to provide correlation among several infrastructure elements. These elements can begin to recognize patterns, predict the optimal configuration and provide advice on what course of action the administrator should take.
4. As these technologies improve and as people become more comfortable with the advice and predictive power of these systems, they can progress to the adaptive level. The IT environment can automatically take actions based on the available information and the knowledge of what is happening in the environment.
5. To get to the fully autonomic level the IT infrastructure operation is governed by business policies and objectives. Users interact with autonomic technology tools to monitor business processes, alter objectives, or both.

The following sections discuss the autonomic computing levels for each autonomic characteristic—self-configuring, self-healing, self-optimizing, and self-protecting. This can help you determine your current level of readiness,

assess the capabilities of current tools and evaluate it within the context of a longer-term view.

Self-configuring

An enterprise can greatly increase its responsiveness to both employees and customers with a self-configuring IT environment. With the ability to dynamically configure itself on the fly, an IT infrastructure can adapt immediately—and with minimal human intervention—to the deployment of new components or changes in the IT environment. For example, an e-business retailer dealing with seasonal workload peaks during the holiday shopping season or increased business for a particular event can use a self-configuring IT infrastructure to reassign servers from under-utilized pools to overutilized ones to match shifting workloads. Tivoli software management tools from IBM can allow you to provision a wide range of resources, including systems, applications, users and access privileges, and physical and logical storage. Monitoring and event correlation tools can help determine when changes in the IT infrastructure warrant reconfiguration actions. These tools can allow you to reconfigure your IT environment within minutes or hours rather than in days or weeks.

IBM has defined five implementation levels for a self-configuring IT infrastructure, based on the major capabilities that should ultimately exist for true autonomic functionality (see Figure A-37 on page 371).

Basic Level 1	Managed Level 2	Predictive Level 3	Adaptive Level 4	Autonomic Level 5
Definition <ul style="list-style-type: none">● Reliance on system reports, product documentation and manual actions to configure, optimize, fix and protect individual IT components● Human intervention required to perform any action Value <p>Single systems configured and administered</p>	Definition <ul style="list-style-type: none">● Software provides monitoring, consolidation, facilitation and automation of IT tasks● Clusters may be used to balance workload● Analysis and actions performed by IT staff Value <p>Helps increase administrator productivity through fast system installation</p>	Definition <ul style="list-style-type: none">● Individual IT components and system tools monitor and analyze changes● User and role-based reallocation of resources● Capacity is adjusted at the resource level to help provide transactional performance Value <p>Helps improve administrative productivity through role-based provisioning</p>	Definition <ul style="list-style-type: none">● IT components individually and collectively monitor, analyze and take action with minimal human intervention● IT policy determines capacity reallocation across specific resource types Value <p>Helps increase system availability and reduces human intervention through event-driven automated provisioning</p>	Definition <ul style="list-style-type: none">● IT components collectively and automatically managed by business rules and policies established in the system● Resources discovered and capacity reallocated Value <p>Helps achieve business resource optimization to meet service-level objectives and business priorities</p>

Figure A-37 Five implementation levels for a self-configuring IT infrastructure

- ▶ **Level 1: Basic**—The focus is on the ability to deploy, configure and change an individual system component, including system hardware configuration, storage hardware configuration, communication configuration and operating system configuration. Basic resource-specific tools are used to perform configuration actions. Configuration of multiple resources is done by logging on to each resource admin tool separately to perform the configuration action.
- ▶ **Level 2: Managed**—The focus is on the ability to deploy and manage change to an aggregated group of systems. This includes managing multiple systems and system images, moving systems in and out of clusters, deploying applications to groups of machines, managing groups of users and moving storage in and out of storage networks. The concept of virtualized storage is introduced and the ability to monitor the collective system health and storage components to make decisions about how they might need to be reallocated and reconfigured is assessed.
- ▶ **Level 3: Predictive**—The notion of managing based on role is introduced, including user role and system role, so that configurations can be appropriately tailored for their use. Configuration sensing (for example, inventory scanning) and runtime monitoring information is used to determine when corrective actions need to be taken. The administrator can initiate corrective actions based on system recommendation.
- ▶ **Level 4: Adaptive**—The focus is on dynamically managing the configuration of the environment by leveraging sophisticated correlation and automation. The key notion is that the reconfiguration happens automatically and the IT infrastructure adjusts itself based on overall configuration health and role changes.
- ▶ **Level 5: Autonomic**—Reconfiguration actions are taken within the context of overall business policies and priorities. Business impacts are assessed to determine the appropriate reconfiguration actions. It also includes the ability to provision proactively and anticipate issues that might jeopardize service levels before breaches actually occur.

Self-configuring capabilities enabled by Tivoli software products

Tivoli software products that can be used to implement a self-configuring environment include:

- ▶ **Tivoli Configuration Manager**—Tivoli Configuration Manager configures automatically to rapidly changing environments. It provides an inventory scanning engine and a state management engine that can sense when software on a target machine is out-of-synch with a reference model for that class of machine. It can automatically create a customized deployment plan for each target and sequence the installation of software in the right order.
- ▶ **Tivoli Identity Manager**—Tivoli Identity Manager automates user lifecycle management and integrates with HR and native repositories. It uses

automated role-based provisioning for account creation. The provisioning system communicates directly with access control systems to help create accounts, supply user information and passwords and define account entitlements.

- ▶ **Tivoli Storage Manager**—Tivoli Storage Manager provides self-configuring capabilities to perform tasks such as automatically identifying and loading the appropriate drivers for the storage devices connected to the server. Configuration and policy information can be defined once at a Tivoli Storage Manager configuration server and then propagated to a number of managed Tivoli Storage Manager servers. Policies and internal automation allow automatic extension of the server database, recovery log, or both when administrator-defined thresholds are reached.

Self-healing

A self-healing IT infrastructure can detect improper operation of systems, transactions and business processes (either predictively or reactively) and then initiate corrective action without disrupting users. Corrective action could mean that a component is altered or other components are altered to accept its workload. Day-to-day operations do not falter or fail because of events at the component level. The Tivoli software availability management portfolio from IBM provides tools to help customers monitor the health and performance of their IT infrastructure. These tools help allow monitoring of multiple metrics from a heterogeneous collection of resources and provide the ability to perform filtering, correlation and analysis. Based on the analysis, automated actions can be taken to cure problems even before they occur. Autonomic capabilities are provided at multiple levels to allow customers to understand business impacts and proactively manage the availability of the IT infrastructure workbench tools allow integration of third-party applications.

IBM has defined five implementation levels for self healing and availability management, based on the major capabilities that should ultimately exist for true autonomic functionality (see Figure A-38).

Basic Level 1	Managed Level 2	Predictive Level 3	Adaptive Level 4	Autonomic Level 5
<p>Definition</p> <ul style="list-style-type: none"> • Local consoles on each system used for administration • Manual reading of logs is the primary method for problem debug <p>Value</p> <p>Helps minimize capital expenditures in the IT cost center</p>	<p>Definition</p> <ul style="list-style-type: none"> • Centrally monitor and report on IT resources • Event filtering, correlation and management products used centrally • Problems fixed manually by skilled administrators <p>Value</p> <p>Helps increase ROI of IT resources and IT services</p>	<p>Definition</p> <ul style="list-style-type: none"> • IT resource contribution to transactional delays monitored and projected • Business impact (IT services impact) of resource problems understood • Helps automate corrective actions for routine problems <p>Value</p> <p>IT resources managed in context of business impact</p>	<p>Definition</p> <ul style="list-style-type: none"> • Policy-based, desired-state driven to return infrastructure to functioning state • Problem trends identified and automatically corrected <p>Value</p> <p>Helps increase business availability</p>	<p>Definition</p> <ul style="list-style-type: none"> • Dynamic reprovisioning of IT infrastructure to accommodate resource failures • Business continuity automatically maintained <p>Value</p> <p>Helps maximize business impact of IT services</p>

Figure A-38 Five implementation levels for self healing and availability management

- ▶ **Level 1: Basic**—Systems administration and problem management is accomplished by using significant human processing power. Availability of systems is addressed in a reactive way. IT staff learns of problems from customers complaining about lack of service. Problem determination, correlation and cures are accomplished with a great deal of human intervention. Highly skilled IT staff are needed to debug problems.
- ▶ **Level 2: Managed**—The focus is on the ability to collect and view availability information from remote locations. Many resources may be located outside the data center, perhaps in branch offices. Error logs can be accessed remotely. IT has deployed a set of monitoring tools to report on availability to a central location. Multiple system and network events can be filtered or manually correlated to identify the root cause of problems. Problems are fixed by skilled administrators.
- ▶ **Level 3: Predictive**—IT has granular views into IT systems to accurately pinpoint the cause of outages. Complex, multiple metric collection is now possible, instead of single metrics. Filtering is now advanced and tied to correlation engines, allowing improved root cause problem determination to take place. Automated corrective actions are taken to known problems. These capabilities help customers prioritize which problem to repair first, based on the business impact of the outage.
- ▶ **Level 4: Adaptive**—Systems can automatically discover, diagnose and fix problems on multiple monitored resources (operating system, application, middleware) across multiple monitored systems. The IT infrastructure availability is maintained automatically to keep in tune with predefined desired states. Outages don't bring down the system; it dynamically adapts to

outages until repairs can be made to maintain service levels. For example, thresholds are temporarily raised to account for added workload.

- ▶ **Level 5: Autonomic**—Problem determination and diagnosis depend on sophisticated knowledge already encoded in the system about components and their relationships. The inference capability allows the system to automatically figure out corrective actions within the right business context. For example, if a particular outage cannot be contained with available resources, lower-priority business applications may be shut down or run with degraded quality of service to keep higher priority business applications functioning.

Self-healing capabilities enabled by Tivoli software products

Tivoli software products that can be used to implement a self-healing environment include:

- ▶ **IBM Tivoli Enterprise™ Console**—Tivoli Enterprise Console® collates error reports, derives root cause and initiates corrective actions. The event server and correlation engine help allow cross-resource correlation of events observed from hardware, applications and network devices throughout an enterprise. Events from multiple resources can be analyzed in real time to automatically highlight the critical problems that merit attention versus the misleading symptoms and effects. After a problem is highlighted, the system takes self-healing actions by responding automatically when possible or efficiently guiding the support staff to the appropriate response.

- ▶ **IBM Tivoli Switch Analyzer**—Tivoli Switch Analyzer correlates network device errors to the root cause without user intervention. It is a Layer 2 switch network management solution that provides automated Layer 2 discovery. It identifies the relationship between devices, including Layer 2 and Layer 3 devices, and identifies the root cause of a problem without human intervention. During a network event storm it can filter out extraneous events to correlate the true cause of the problem.
- ▶ **IBM Tivoli NetView®**—Tivoli NetView helps enable self-healing by discovering TCP/IP networks, displaying network topologies, correlating and managing events and SNMP traps, monitoring network health, and gathering performance data. Router fault isolation technology quickly identifies and focuses on the root cause of a network error and initiates corrective actions.
- ▶ **Tivoli Business Systems Manager**—Tivoli Business Systems Manager collects real time operating data from distributed application components and resources across the enterprise and provides a comprehensive view of the IT infrastructure components that make up different business solutions. It contains technologies that analyze how an outage would affect a line of business, critical business process, or service level agreement (SLA).
- ▶ **Tivoli Systems Automation S/390**—Tivoli Systems Automation S/390 manages real time problems in the context of an enterprise's business priorities. It provides monitoring and management of critical system resources such as processors, subsystems, and Sysplex Timer® and coupling facilities. It supports self-healing by providing mechanisms to reconfigure a processor's partitions, perform power-on reset on IML processors, IPL operating systems (even automatically), investigate and respond to I/O configuration errors, and restart and stop applications if failures occur.
- ▶ **IBM Tivoli Risk Manager**—Tivoli Risk Manager enables self-healing by assessing potential security threats and automating responses, such as server reconfiguration, security patch deployment, and account revocation. This helps enable system administrators who are not security experts to monitor and assess security risks in real time with a high degree of integrity and confidence across an organization's multiple security checkpoints. This product contains technology from IBM Research.
- ▶ **IBM Tivoli Monitoring for Applications, IBM Tivoli Monitoring for Databases, and IBM Tivoli Monitoring for Middleware**—This family of products minimizes vulnerability by discovering, diagnosing and reacting to disruptions automatically. It provides monitoring solutions and a local automation capability through a set of Proactive Analysis Components. A sophisticated resource model engine allows for local filtering of monitored data, raising events when specific conditions are met. Local rules can be encoded to take immediate corrective action, providing automatic recovery for server failures.

- ▶ **Tivoli Storage Resource Manager**—Tivoli Storage Resource Manager automatically identifies potential problems and executes policy-based actions to help prevent or resolve storage issues, minimize storage costs and provide application availability. It can scan and discover storage resources in the IT environment. It supports policy-based automation for the allocation of storage quotas and storage space, monitors file systems and provides reports on capacity and storage asset utilization.

Self-optimizing

Self-optimization is the ability of the IT infrastructure to efficiently maximize resource allocation and utilization to provide Quality of Service for both system users and their customers. In the near term self-optimization primarily addresses the complexity of managing system performance. In the long term self-optimizing software applications may learn from experience and proactively tune themselves in an overall business objective context. Workload management uses self-optimizing technology to help optimize hardware and software use and verify that service level goals are being met. Predictive analysis tools provide views into performance trends, allowing proactive action to be taken to help optimize the IT infrastructure before critical thresholds are exceeded.

IBM has defined five implementation levels for a self-optimizing IT infrastructure that can optimize workloads and transaction performance across multiple resources (see Figure A-39).

Basic Level 1	Managed Level 2	Predictive Level 3	Adaptive Level 4	Autonomic Level 5
Definition <ul style="list-style-type: none"> • Monitor and report on individual IT resource performance in disparate formats • Manually optimize the performance of individual IT resources Value <p>Can minimize capital expenditures in the IT cost center</p>	Definition <ul style="list-style-type: none"> • Monitor and report on end-user performance (transactions) • Centralized, comprehensive performance visibility • Manually optimize groups of linked resources Value <p>Helps increase ROI of IT resources and IT services</p>	Definition <ul style="list-style-type: none"> • Service-level agreements and priorities are committed and managed • Realtime service-level visibility and projected future-level visibility • Analysis and recommendations for manual tuning Value <p>Predictable performance and value of IT services</p>	Definition <ul style="list-style-type: none"> • Policy-based desired-state driven • Service-level priorities drive automatic resource (re)configuration and shift workload to meet objectives • Automated closed-loop resource tuning Value <p>Flexibility, fast deployment of new IT services</p>	Definition <ul style="list-style-type: none"> • Dynamic reprovisioning of aspects of IT infrastructure to meet shifting workloads and service-level commitments • Automated closed-loop transaction tuning Value <p>Helps maximize business impact of IT services</p>

Figure A-39 Five implementation levels for a self-optimizing IT infrastructure that can optimize workloads and transaction performance across multiple resources

- ▶ **Level 1: Basic**—Individual resources provide point data regarding individual component performance or utilization, allowing users a simple view of how

workload affects a single system. Basic tools allow dynamic viewing of components, but comprehensive views of system performance are still put together manually by looking at multiple local views and reports available with the resource-specific tools.

- ▶ **Level 2: Managed**—Management tools allow information on resource utilization and performance to be gathered and collected in a central location. Simple, comprehensive transaction views are possible using techniques such as round-trip measurements, synthetic transactions or end-user client-capture capabilities. Many resources in the middle of a transaction are invisible or not instrumented, and many resources are often located outside the data center—perhaps in branch offices or other locations. Optimizing the IT components is still done manually and with trial and error.
- ▶ **Level 3: Predictive**—Tools now provide value by creating detailed, comprehensive transaction views and can break down the composite view of the transaction across the resource elements. Resources can be grouped by transaction types, service levels can be monitored and automated tools provide notifications of impending violations—allowing manual reconfiguration of the IT environment. Predictive tools can perform trend analysis on historical data and provide recommendations.
- ▶ **Level 4: Adaptive**—Instrumentation is now available on the composite resources to allow changes of status and automated balancing of work when overload or underload conditions exist across resources in the environment. This level of control provides users with the ability to manage comprehensive performance and effectively meet SLAs.
- ▶ **Level 5: Autonomic**—Workload balancing and transaction optimization is done within the business context. Business trade-offs are expressed in machine-processable format, allowing IT management tools to dynamically reallocate resources based on varying business needs. Automated tuning of servers, storage, and networks takes place to maintain quality of service for high-priority business applications.

Self-optimizing capabilities enabled by Tivoli software products

Tivoli software products that can be used to implement a self-optimizing environment include:

- ▶ **Tivoli Service Level Advisor**—Tivoli Service Level Advisor helps prevent SLA breaches with predictive capabilities. It performs trend analysis based on historical performance data from Tivoli Enterprise Data Warehouse and can predict when critical thresholds could be exceeded in the future. By sending an event to Tivoli Enterprise Console, self-optimizing actions can be taken to help prevent the problem from occurring.
- ▶ **IBM Tivoli Workload Scheduler for Applications**—Tivoli Workload Scheduler for Applications automates, monitors, and controls the flow of work

through the IT infrastructure on both local and remote systems. It can automate, plan, and control the processing of these workloads within the context of business policies. It uses sophisticated algorithms to maximize throughput and help optimize resource usage.

- ▶ **Tivoli Business Systems Manager**—Tivoli Business Systems Manager enables optimization of IT problem repairs based on business impact of outages. It collects real time operating data from distributed application components and resources across the enterprise and provides a comprehensive view of the IT infrastructure components that make up different business solutions. It works with Tivoli Enterprise Console to enable self-optimizing actions to help prevent poor performance from affecting a line of business, critical business process, or SLA.
- ▶ **Tivoli Storage Manager**—Tivoli Storage Manager supports Adaptive Differencing technology to help optimize resource usage for backup. With Adaptive Differencing, the *backuparchive* client dynamically determines an efficient approach for creating backup copies of just the changed bytes, changed blocks or changed files, delivering improved backup performance over dialup connections. These technologies allow just the minimum amount of data to be moved to backup, helping optimize network bandwidth, tape usage, and management overhead.
- ▶ **Tivoli Monitoring for Transaction Performance**—Tivoli Monitoring for Transaction Performance helps customers tune their IT environments to meet predefined service level objectives. It enables organizations to monitor the performance and availability of their e-business and enterprise transactions to provide a positive customer experience. It integrates with the Tivoli Enterprise Console environment for alerting and proactive management, helping enable optimization of resource usage from a transactional perspective.
- ▶ **IBM Tivoli Analyzer for Lotus Domino**—Tivoli Analyzer for Lotus Domino contains a Proactive Analysis Component that allows administrators to verify the availability and optimal performance of Lotus Domino servers. It provides intelligent server health monitoring and expert recommendations to correct problems.

Self-protecting

A self-protecting IT environment can take appropriate actions automatically to make itself less vulnerable to attacks on its runtime infrastructure and business data. These attacks can take the form of unauthorized access and use, malicious viruses that can format hard drives and destroy business data, and denial-of-service attacks that can cripple critical business applications.

A combination of security management tools and storage management tools are necessary to deal with these threats. Security management tools can help businesses consistently enforce security and privacy policies, help reduce

overall security administration costs, and help increase employee productivity and customer satisfaction. Critical configuration changes and access-control changes should only occur with the right approvals. Tools should detect violations of security policy, and if necessary, automated actions should be taken to minimize risk to IT assets. Tivoli software storage management tools help enable businesses to automatically and efficiently back up and protect business data. Autonomic security and storage solutions provide administrators with a way to create policy definitions and express event correlation and automation knowledge.

IBM has defined five implementation levels for a self-protecting IT infrastructure (see Figure A-40).

Basic Level 1	Managed Level 2	Predictive Level 3	Adaptive Level 4	Autonomic Level 5
Definition <ul style="list-style-type: none"> Security controls are implemented on each device (individual formats) Basic infrastructure tools are deployed to protect corporate assets Value <p>Can minimize capital expenditures in the IT cost center</p>	Definition <ul style="list-style-type: none"> Security configurations for devices are managed centrally Intrusion sensor devices are deployed throughout the infrastructure Value <p>Helps improve ability to protect corporate assets</p>	Definition <ul style="list-style-type: none"> Application security is managed according to common security policy Correlation of intrusion events to identify real threats from normal business activity Consolidated sign-on Value <p>Helps improve application security with a common security infrastructure</p>	Definition <ul style="list-style-type: none"> Security access is granted to users and enforced dynamically based on user-policy changes Infrastructure intrusion-detection tool configuration adapts to IT security policy and threats Value <p>Security infrastructure responds with minimal human intervention</p>	Definition <ul style="list-style-type: none"> e-business security context Self-collaborating systems for detection, verification and reconfiguration Learned policies for future protection Value <p>Helps integrate perimeter and application security to business policy</p>

Figure A-40 Five implementation levels for a self-protecting IT infrastructure

- ▶ **Level 1: Basic**—Localized security configuration requires administrators to configure each component independently and manually track changes. Local backup and recovery tools are used to protect data. Audit reports are on a per-machine basis. A great deal of human intervention is required to protect the runtime and business data.

- ▶ **Level 2: Managed**—Management tools are used to centralize security administration, allowing the centralized creation of user IDs and controlling access privileges to resources. Intrusion sensing and auditing tools are used to collect data about intrusion attempts. These are manually reviewed, and corrective actions are taken to protect against future attacks. Centralized backup and recovery tools provide an incremental backup capability across multiple resources.
- ▶ **Level 3: Predictive**—Security policies can be consistently administered across the enterprise manually, using enterprise-wide security management tools. IDs and access privileges are coordinated across multiple applications and can be consistently revoked if necessary. Perimeter sensors can detect security violations and correlate them to detect attacks. Security tools provide recommendations for corrective action.
- ▶ **Level 4: Adaptive**—Security management focuses on advanced automation, including automatically enabling new users and disabling IDs for those who leave. It automatically grants access to systems and applications needed to do a new job while disabling access to systems associated with the old job. If access control violations and intrusions are detected, automatic reconfiguration actions are initiated to help quarantine systems and disable access to IDs.
- ▶ **Level 5: Autonomic**—The focus is on learning systems and systems that can adapt lower-level resource policies in response to higher-level business policy. Collaboration across system components makes it possible to reconfigure systems on the fly, automatically apply security patches when necessary, modify intrusion monitoring levels based on business needs, and adapt policies to help prevent future problems based on past history.

Self-protecting capabilities enabled by Tivoli software products

Tivoli software products that can be used to implement a self-protecting environment include:

- ▶ **Tivoli Storage Manager**—Tivoli Storage Manager self-protects by automating backup and archival of enterprise data across heterogeneous storage environments. Scaling to protect thousands of computers running a dozen operating system platforms, its intelligent data movement and store techniques and comprehensive automation help reduce administration costs and increase service levels.
- ▶ **Tivoli Access Manager**—The Tivoli Access Manager family of products self-protects by helping prevent unauthorized access and using a single security policy server to enforce security across multiple file types, applications, devices, operating systems, and protocols. It supports a broad range of user authentication methods, including Web single sign-on, and has

- the ability to control access to many types of resources for authenticated users.
- ▶ **Tivoli Identity Manager**—Tivoli Identity Manager self-protects by centralizing identity management, integrating automated workflow with business processes, and leveraging self-service interfaces to increase productivity.
- ▶ **Tivoli Risk Manager**—Tivoli Risk Manager provides system-wide self-protection by assessing potential security threats and automating responses, such as server reconfiguration, security patch deployment, and account revocation. It collects security information from firewalls, intrusion detectors, vulnerability scanning tools, and other security checkpoints. It simplifies and correlates the vast number of events and alerts generated by numerous security point products and quickly identifies the real security threats to help administrators respond with adaptive security measures.
- ▶ **IBM Tivoli Privacy Manager for e-business**—Tivoli Privacy Manager for e-business self-protects by automating many privacy-compliance activities, simplifying the incorporation, monitoring, and enforcement of privacy policy into business processes. It can record end user's "opt-in" and "opt-out" choices according to the policy, can be used to monitor and enforce access according to the privacy policy, and can create audit trail reports.

Summary

Companies want and need to reduce their IT costs, simplify management of their IT resources, realize a fast return on their IT investment, and provide high levels of availability, performance, security, and asset utilization. Autonomic computing helps address these issues. IBM is a leader in the evolution to autonomic computing and offers integrated systems management solutions for resource management, transaction-oriented management, and business-solution management that span the four autonomic computing disciplines of self-configuring, self-healing, self-optimizing, and self-protecting.



B

Using WebSphere Portal

Note: The following comes from *Guide to WebSphere Portal 4.2—A technical discussion on portals* (February 2003).

Portals serve as a simple, unified access point to Web applications. Portals also do much more—they provide valuable functions like security, search, collaboration, and workflow. A portal delivers integrated content and applications, plus a unified, collaborative workplace. Indeed, portals are the next-generation desktop, delivering e-business applications over the Web to all kinds of client devices.

A complete portal solution should provide users with convenient access to everything they need to get their tasks done—any time, anywhere, in a secure manner. IBM's vision is that portals are the key to reach and user experience of an e-business application. That is, portals provide the tools and user interface to access information and applications, and to manage the selection and personalization of content.

The WebSphere Platform IBM WebSphere Portal is part of the WebSphere software platform. The platform is organized into three areas of functionality:

1. **Foundation and tools** for building, running, and deploying applications. WebSphere Application Server, MQ messaging, and state-of-the-art development tools form a solid base for the platform. The foundation and tools provide the Internet expertise you need, enable you to build and use

Web services, and link you to a greater technical community of developers and other WebSphere users.

2. **Business integration** for integrating internal business processes, including processes that involve business partners. WebSphere offerings such as WebSphere Business Integrator make it easy for your company to implement applications and business processes, including supply-chain management and the integration of existing processes with the Web.
3. **Reach and user experience** for personalizing Web-based content and making it accessible to any device. These WebSphere products fine-tune your users' experience and provide broad access for your customers, employees, business partners, and remote branch offices.

WebSphere Portal leads the Reach and User Experience part of the WebSphere Platform. It provides an extensible framework for interacting with enterprise applications, content, people, and processes. Self-service features allow end users to personalize and organize their own view of the portal in order to manage their own profiles and to publish and share documents with their colleagues.

WebSphere Portal provides additional services such as single sign-on, security, Web content publishing, search and personalization, collaboration services, enterprise application integration, support for mobile devices, and site analytics.

B.1 WebSphere Portal architecture

WebSphere Portal is the industry's most comprehensive portal solution and represents the *de facto* standard e-business architecture. WebSphere Portal integrates both IBM and business partner technologies to realize this architecture. IBM is also extending its portal offering to create additional products that deliver highly personalized and context-sensitive applications, accessible from any device at any time.

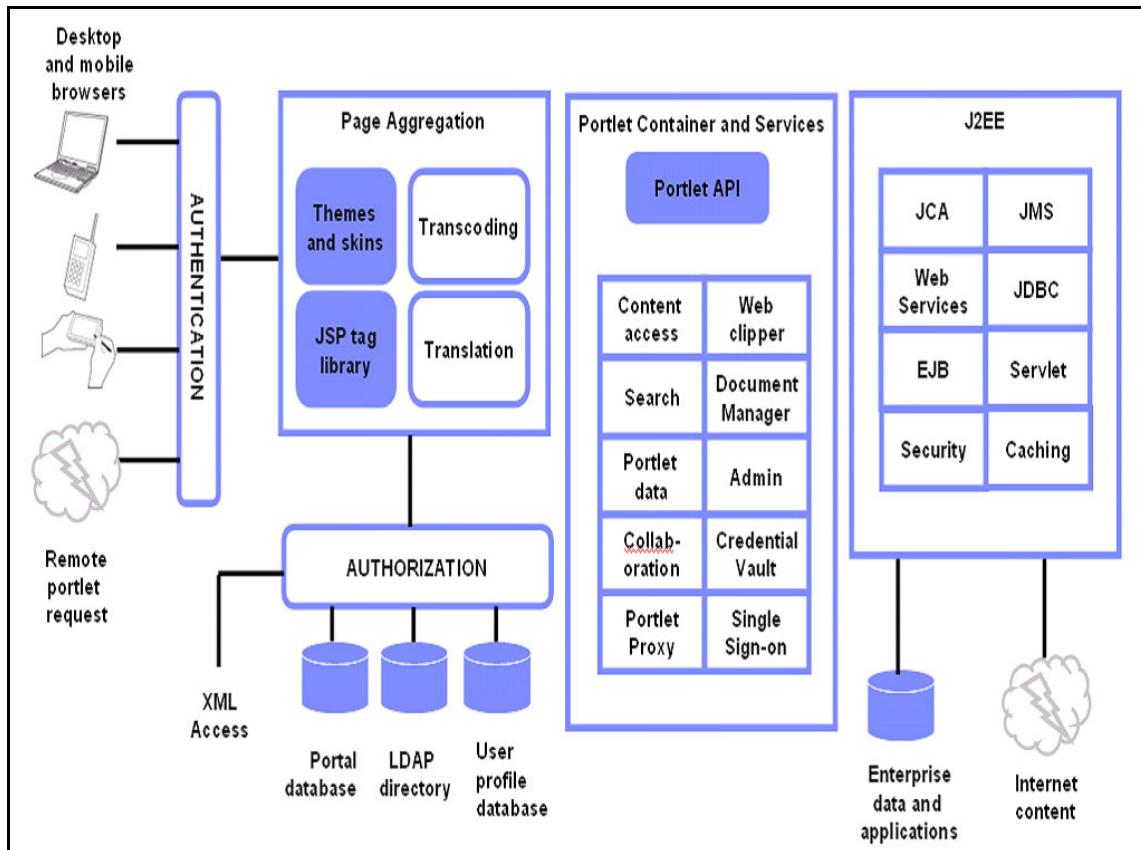


Figure B-1 WebSphere Portal architecture

As e-business applications enter the on demand era, WebSphere Portal leads the way with its concepts of delegated administration, cascading page layouts, portal federation through Web services, advanced portlet application concepts, business process integration, knowledge management, and advanced personalization. In complementary offerings, additional pervasive computing functions are enabled, such as intelligent notification, offline browsing, and data synchronization.

WebSphere Portal is available in several editions, each designed to provide the infrastructure you need to build and deploy highly scalable portals. All three offerings share a common framework (the portal server) plus additional products and services. The portal server provides common services such as application connectivity, integration, administration, and presentation that are required across portal environments.

B.1.1 Portlets

Portlets are the heart of a portal. The term *portlet* refers to a small portal application, usually depicted as a small box in the Web page. Portlets are reusable components that provide access to applications, Web-based content, and other resources. Web pages, Web services, applications, and syndicated content feeds can be accessed through portlets. Companies can create their own portlets or select from a catalog of portlets created by IBM and its business partners.

Any particular portlet is developed, deployed, managed, and displayed independent of other portlets. Administrators and end users create personalized portal pages by choosing and arranging portlets. The portal server already includes a rich set of standard portlets for displaying syndicated content, performing XML transformation—accessing existing Web pages, Lotus Notes and Microsoft Exchange productivity applications, Lotus Instant Messaging, and Lotus Team Workplace team rooms.⁹⁸

Portlet applications

Portlets are more than simple views of existing Web content. A portlet is a complete application, following a standard model-view-controller design. Portlets have multiple states and view modes, plus event and messaging capabilities.

Portlets run inside the portlet container of a portal server, similar to a servlet running on an application server. The portlet container provides a runtime environment in which portlets are instantiated, used, and finally destroyed. Portlets rely on the portal infrastructure to access user profile information, participate in window and action events, communicate with other portlets, access remote content, look up credentials, and store persistent data.

Portlet modes

Portlet modes allow a portlet to display a different user interface, depending on the task required of the portlet. A portlet has several modes of display, which can be invoked by icons on the portlet title bar (view, help, edit, and configure).

Portlet API

Portlets are a special subclass of *HttpServlet*, with properties that allow them to easily plug into and run in the portal server. Portlets are assembled into a larger portal page, with multiple occurrences of the same portlet displaying different data for each user.

⁹⁸ An extensive third-party partner program makes many more portlets available for IBMers from the portlet catalog download site at
http://www-306.ibm.com/software/info1/websphere/index.jsp?tab=products/portal&S_TACT=103BE_W01

Standards

As portals continue to evolve into the new desktop and integration standard, IBM is leading efforts to standardize the application programming interfaces between portals and other applications. In particular, the Java Community Process (JCP) and the Organization for the Advancement of Structured Information Standards (OASIS) are working cooperatively to standardize the Java and XML technology needed to link portals to disparate applications.

Struts

Struts is a Jakarta open source project that provides a framework based on the Model-View-Controller (MVC) pattern. It allows developers to efficiently separate the application's business logic from its presentation. Struts enforces sequence of pages and actions and provides form validation functions.

The latest release of WebSphere Portal adds support for using the Struts 1.1 framework to build portlets.

Portlet cooperation

The portal server provides a mechanism for portlets to communicate with each other, exchanging data or other messages.

Brokered cooperation

Brokered cooperation allows independently developed portlets to exchange information. Portlets that exchange data in response to a user action are called *click-to-action* portlets. Click-to-action automatically matches the portlet information sources and possible actions based on their data type compatibility. A unique advantage of click-to-action is that it is designed to work in different browsers, making it more accessible to users.

Discoverable services

The Portlet API provides an interface to enable dynamic discovery of available services. Each service is registered in a portal configuration file and is accessed from the *PortletContext.getService()* method, which looks up the factory for the service, creates the service, and returns it to the portlet. This makes services available to all portlets without having to package the service code with the portlet. The implementation of such a service can be exchanged or enhanced transparently, without affecting the portlet.

The portal server provides discoverable services for its credential vault, for managing persistent TCP/IP connections, and for managing the portal's content repository. New services such as search, location, notification, content access, or mail services can be implemented by portal developers.

Tools

Portlets can be grouped together in a portlet application. Portlet applications are distributed and deployed using Web archive (WAR) files. There are portlet-specific extensions to the standard Web application deployment descriptor.

WebSphere Studio Application Developer provides an excellent development, test, and debug environment for portlet applications. You can implement the Java classes for portlet classes and also test and debug the Java code. Studio also provides tools for creating Java Server Pages, HTML pages, images, and other related portal resources. The portal toolkit (downloaded from the Web) provides plug-ins to help build the portlet deployment descriptor, package it into a WAR file, and then easily deploy it to the runtime portal server.

B.1.2 Content and search

WebSphere Portal meets the portal's content delivery needs by supporting syndicated content, by integration with Web content management systems, and by providing built-in content and search portlets.

Syndicated content

A key concept related to portal technology is syndication, which is about delivering fresh, personalized, and filtered content and services from multiple sources to subscribers. Typically, the content is related to news, finance, and entertainment. Portal partners include popular content providers such as Pinnacor, Financial Times, Moreover, YellowBrix, Hoovers, Factiva, NewsEdge, MediaApps, and DataMonitor.

Companies are embracing syndication concepts and standards to automate the publishing of electronic catalogs and other internal information, and to make this information available to workers through enterprise portals. A popular and useful format for syndicated news and entertainment content is Rich Site Summary (RSS). Content can be published directly from the content management system into Rich Site Summary and Open Content Syndication (OCS) channels, where it can easily be displayed by the portal server's built-in RSS portlet. This self-syndication concept defines a procedure for editing, managing, and publishing your own sources of content.

Web content management

Web content management deals with creating, approving, and publishing Web content from content creators to Web servers. The steps of this process include defining content types, roles, publication options, destination specifications, and workflow processes. There are many content management vendors in the marketplace today, including Interwoven, Vignette, Documentum, FatWire,

Stellant, and IBM Lotus Workplace Web Content Management. Although each of these products works differently, generally they are designed to create, maintain, and publish collections of documents that can be made available to users via the portal.

WebSphere Portal includes integration kits that illustrate specific steps of how to publish RSS content from several of the Web content management products. These are available through the portlet catalog. Content contribution and approval operations of the Web content management system can also be accessed through portlets that are provided by their respective companies. These portlets provide a user interface into various aspects of the content management process—such as content submission, workflow management, content approval, and even staging or publishing.

Web content publishing

The Web Content Publisher is a tool intended for users that need to contribute content to a Web site on a regular or occasional basis. This includes business users creating template-driven content such as press releases or product information, and graphic artists creating and editing artwork using their favorite tools. Content Publisher supports content contribution via templates or forms, as well the contribution of files such as images, HTML, or JSPs, which are created and edited using popular tools such as HomePage Builder, Dreamweaver, FrontPage, PhotoShop, or Word.

Workflow

Web Content Publisher manages the task lists of each user, maintains access control over what content each user can see or change, and coordinates the approval and publishing process when the content is ready.

Digital content libraries

For managing and searching large collections of digital content, the WebSphere Portal Experience offering includes IBM Content Manager. Additional search portlets from leading search vendors such as Verity, Inktomi, and Autonomy are also available and can be found on the portlet catalog.

Search

WebSphere Portal provides integrated text search capabilities, including a search portlet, a crawler, and a document indexer. The search service can search local documents as well as Internet content. The portal server's built-in search engine is optimized for full-text searching of small and medium-sized collections, where precision is essential. It efficiently applies state-of-the-art search algorithms, producing high-quality search results.

The search engine supports free-text queries, with query assistance and query word completion. Search queries use advanced query operators (+ or -) to indicate keywords that must or must not be in the document. The search engine can search documents in any language, and also supports synonyms and stop word lists. Search results include document summarization and search results clustering.

Federated search

WebSphere Portal also integrates several other search technologies. Portlets using IBM Lotus Extended Search and DB2 Information Integrator can access and aggregate other search engines and indexes in a distributed fashion. Customers seeking support for large document collections or for searching a wide range of document types and data sources should consider using Lotus Extended Search⁹⁹ or IBM's Content Manager.

Extended search

Lotus Extended Search (ES) provides distributed, heterogeneous searching across Domino servers, databases, and the Internet, without the user having to know the details of these various systems. The result is single point of access to a variety of data sources without requiring a new, central index. ES can search and retrieve documents from repositories that include Lotus Notes 4.X and 5.X, Domino.doc, and R5 Domain Index. ES also searches external sources such as Microsoft Index Server and Site Server, LDAP-compliant directories, 18 popular Web search sites and News sites, commercial content providers, and relational databases such as IBM DB2, Oracle, Sybase, MS SQL Server, and other ODBC compliant databases. Results can be ranked by relevancy over multiple data stores.

Advanced search and document processing

IBM Content Manager can manage data access across multiple sources such as content management repositories, e-mail systems, relational databases, file systems, and Web sites (both intranet and Internet). The developer's interface for working with IBM Content Manager is called *DB2 Information Integrator for Content*. It integrates data sources across the enterprise through a unified set of APIs to simplify programming and to speed development and deployment while providing an interface layer that isolates portal applications from changes to underlying data repositories.

Documents can be full-text indexed and searched using the Information Integrator's crawler and text search features. Many different document formats are supported in addition to standard markup text such as HTML and XML. Documents can be categorized into taxonomies, enabling search by category.

⁹⁹ Included with the WebSphere Portal Extend offering.

APIs are provided for capturing and storing other metadata about documents. Information Integrator has connectors for a variety of repositories provided by IBM, Lotus, and other vendors such as Documentum and Filenet. Federated searches can be applied across multiple such repositories and can exercise searching based on metadata, full text, and other specialized search properties such as Query by Image Content (QBIC®)—depending on the search services enabled for each repository. The Text Analysis features support creating full-text indices and subsequent searching across all the text portions of the content sources configured for use in the portal. Sources can be accessed for indexing via the Web crawler or a metadata search. Portlets for accessing Information Integrator's advanced and federated search functions are available from the portlet catalog.

B.1.3 Security

With the explosive growth of B2B and B2C Web applications, e-businesses must protect critical information assets from intruders and hackers. Service providers need similar protection when re-hosting e-business content and applications for their customers. Portal applications and resources are protected by:

- ▶ Managing user profiles (*member services*)
- ▶ Verifying user identity (*authentication*)
- ▶ Managing access to back-end applications (*single sign-on*)
- ▶ Enforcing access policies (*authorization*)

Member services

Centralized administration of user identities, credentials, and permissions is desirable in many environments. The portal server includes facilities for defining portal users and managing user access rights. The user and group subsystem includes Web pages where users can register and manage their own account information, administration portlets for managing user accounts and group information, and a repository that stores all the information about portal users. It provides services to create, read, update, and delete users or groups in the repository. User profile information includes general information such as a user's name and ID, plus preference information such as news topics of interest and preferred language. A user can be a member of one or more groups, and groups can contain other groups.

The default set of user profile attributes is based on the *inetOrgPerson* schema, which is supported by most LDAP directories. The user repository can consist of multiple data sources. By default, the repository consists of two data sources—it is a combination of a database and a directory server. The database can be either a DB2 or Oracle database. Additional ones may be supported in the future. Any one of several LDAP directory products are supported, including the

Netscape (iPlanet) Directory Server, Microsoft Active Directory, Novell eDirectory, Lotus Domino, and IBM Directory Server.

Administration

Administration of users and groups can be performed by users themselves (*self care*) or by portal administrators. The portal server includes forms for registering new users as well as administration portlets for updating user and group information.

Authentication

Authentication is the process of establishing a user's identity. Usually, the portal server uses the authentication services provided by WebSphere Application Server. Another option is to use a third-party authentication server (such as Tivoli Access Manager, WebSEAL, or Netegrity SiteMinder) that has a trusted association with the application server.

Identifying the user

Portal server uses form-based authentication, which means that a user is prompted through an HTML form for the user ID and password for authentication when trying to access the portal. The portal server requests the application server to validate the authentication information against a Lightweight Directory Access Protocol (LDAP) user registry.

WebSphere Application Server uses Lightweight Third Party Authentication (LTPA) as the authentication mechanism. A Common Object Request Broker Architecture (CORBA) credential is used to represent authenticated users and their group memberships. When a user tries to access a protected resource, the application server intercepts the request and redirects the request to the login form. This form posts the user ID and password to the portal, which requests the application server to authenticate the user. If the user can be authenticated, a valid CORBA credential is created and an LTPA cookie is stored on the user's machine.

Third-party authentication servers

If your system uses a third-party authentication server, trust must be established between that proxy and WebSphere Application Server. This is done using a Trust Association Interceptor (TAI) module, which converts security information specific to the authentication proxy into a format that can be handled by the application server. The supported authentication mechanism depends on the capabilities of the third-party product.

When a user tries to access the portal, the third-party authentication proxy intercepts the request and challenges that person to authenticate. After a

successful login, the original user request along with additional security information in the request header is passed to the application server. The format and content of this information is vendor-specific. WebSphere Application Server uses the TAI module specific to the third-party product to extract the necessary security information from the request header.

TAI modules for IBM Tivoli Access Manager and Netegrity SiteMinder are packaged with all editions of the portal server. The WebSphere Application Server InfoCenter includes information about creating custom TAI modules for other third-party reverse proxy servers.

Single sign-on

The portal server provides comprehensive single sign-on (SSO) support. Users want to log on only once and be known to the different parts of the portal server with the same consistent user credentials. Users should not be asked to do multiple logons simply because they access different portal applications.

The portal server supports single sign-on realms using WebSphere Application Server as well as authentication proxies. This means that the user needs to log on only once to gain access to all enterprise applications that are installed within the single sign-on realm.

Credential vault

Many portlets must access remote applications that require some form of user authentication. For accessing applications outside the portal's realm, portal server provides a credential vault service that portlets can use to store user ID and password or other credentials for a user login to an application. Portlets can use these on behalf of the user to access remote systems. The credential vault supports either local database storage or IBM Tivoli's Access Manager for secure storage and retrieval of credentials.

For secure transmission of data, portlets can request a secure session (HTTPS) for accessing Web applications.

Persistent connections

Portlets that depend on remote connections require some way of maintaining that connection as users navigate through the portal. The portal provides a persistent back-end connection service that maintains TCP/IP connections across page changes. Some remote applications use forms-based logins and store cookies during the login form processing. The *HttpFormBasedCredential* can be used for handling these form-based logins and will store all the cookies that are returned as a result. For subsequent calls, the portlet can then ask the credential for an authenticated connection. This gives an HTTP connection with

these cookies already set in the header. This way, portlets can maintain persistent and secure back-end connections.

Java security

The portal server implements the Java Authentication and Authorization Service (JAAS) architecture. JAAS provides a means for authenticating subjects and for providing fine-grained access control. JAAS is part of the standard Java security model, giving applications independence from the underlying authentication and authorization mechanisms being used.

JAAS performs login and logout operations using a modular service provider interface. Credentials that are established through the portal server's JAAS login modules include CORBA credentials, user and group distinguished names, user ID and password, and LTPA tokens. In a distributed J2EE environment, portlets can use the JAAS API to access JAAS-enabled back-end applications.

Authorization

After determining the user's identity, the portal server consults locally cached access control lists to determine which pages and portlets a user has permission to access.

The portal server enforces access control to portal assets, including portlets, pages, places, and user groups. The access control lists are stored in the portal's administration database. It is also possible to manage access control for specific resources in an external security manager such IBM Tivoli Access Manager or Netegrity SiteMinder.

Delegated administration

Granting view access to administration portlets is an effective way of delegating certain administrative tasks to other portal users. Those users can simply add the administration portlets to their personal pages and then perform whatever task the portlet is designed to do. This way, the user doesn't have to be given all administrative privileges or added to the portal administrator's group. Each user's administrative abilities are limited to only those tasks covered by the authorized portlets.

B.1.4 Personalizing the portal

Optimizing each user's experience in the portal is one of the goals of WebSphere Portal. To this end, the portal server provides end-user and administrative interfaces for customizing the content of portal pages, as well as the look and layout of the pages. With these tools, users can customize their own pages by selecting portlets and customizing the settings of each one. Users can also

change the page layout and the color scheme if the administrator has decided to allow this.

Customizing pages

Users can have one or more personalized pages, navigating to each one from the home page. Pages are arranged into *places*. In a place, each personalized page can have a different set of portlets. The portlets on a page can be selected by end users or administrators, depending on their access rights for the page.

Cascading portals

Many companies need to build portals that accommodate the needs of a central organization as well as regional and local communities of users. WebSphere Portal supports the needs of companies that want to cascade portal definitions across the organization using the concept of *derived pages*.

Base pages are defined by top-level administrators, who then permit subordinate administrators to further refine the page layouts and content to meet their individual needs. The refinement process can continue for any number of levels until it finally reaches end users. If permitted, end users can customize their own pages by selecting and placing portlets and by changing portlet settings. In order to compute the page that a user finally sees, the portal server merges the page fragments defined by each successive refinement.

A company can achieve both vertical administration (across the company, business segments, locations, and users) and horizontal administration (where administrators provide content that is dedicated for a specific page). Using page locks together with the portal's access control features, administrators can determine which content is *mandatory*, *recommended*, *optional*, or *forbidden*.

Skins and themes

The portal server uses a system of JavaServer Pages (JSP) templates, cascading style sheets, and images to define the look of the portal pages. You can modify these to control any of the visual aspects of the portal, perhaps to add company-specific brand elements or to achieve a different color scheme and visual style.

The system for defining color themes and portal skins has been enhanced to support multiple skins per theme, additional branding elements, navigation styles, and dynamic, browser-independent cascading style sheets. Skins and themes can now be applied to a place rather than being restricted to having only one for the overall portal. Different skins can be applied individually to portlets so that the portal look can be fine-tuned to meet any needs.

Brand elements

All of the visual elements of the portal—including the masthead, the navigation areas, graphics, portlet title areas, and style sheets—can be changed to give the portal a custom look. Standard file formats such as JPEG, GIF, CSS, and JSP files are used for defining the look and the layout of the portal.

Navigation

In this release, the portal's navigation capabilities have been improved significantly. Now, instead of a simple page list for each place, it is possible to create very complex navigation trees with both page links and links to external URLs shown in the navigation area.

Virtual portals

Using different themes for each place, a single installation of the portal server can give the appearance of supporting many “virtual” portals. For example, a company might want to have a different portal for each division or special branding of its B2B portal for each business partner. Each virtual portal can be designed using one or more places, each with its own theme, skins, page layouts, and access permissions.

Universal access

The entire system of page templates, themes, skins, and portlet rendering is fully enabled for internationalization (including double-byte and bi-directional languages) and for accessibility to people with disabilities.

The portal server generates markup that complies with the American Disability Act (ADA) as defined in Section 508 Web Accessibility Standards, and meets the guidelines of the W3C Web Accessibility Initiative.

For globally accessible portals, the portal server will search for and select the proper JSP pages based on the target browser's settings for language and country.

For portals that must support many languages or for those that include rapidly changing portal content, WebSphere Translation Server offers automatic translation technology. This technology is useful for either real time or offline translation of human languages.

WebSphere Translation Server provides quick, inexpensive, convenient “gist” text translations whenever a professional translation is not feasible due to availability, time, or cost. The dictionaries can be tuned so that idiomatic expressions and specialized terminology are interpreted correctly. It supports bi-directional translation of content for English to and from French, Italian, German, Spanish, Japanese, and Chinese (simplified and traditional).

Unidirectional translation of content from English to Korean and Brazilian Portuguese is also supported.

Personalization

WebSphere Portal offerings include the WebSphere Personalization Server, whose purpose is to select content for users based on information in their profiles and on business logic.

WebSphere Personalization provides facilities that allow subject matter experts (SMEs) to select content suited to the unique needs and interests of each site visitor. Web-based tools help companies quickly and easily leverage content created by lines of business (LOBs) and SMEs.

The WebSphere Personalization Server and WebSphere Portal Server share a common user profile and content model. This model is based on the WebSphere resource framework interfaces classes. This means that personalization rules can easily be added to portlets to select portal content and target it to the portal's registered users.

Business experts create the rules for classifying users and selecting content, using Web-based tools. WebSphere Personalization also includes a recommendation engine that provides collaborative filtering capabilities. This uses statistical techniques to identify groups of users with similar interests or behaviors. Inferences can be made about a particular user's interests based on the interests of the other members of the group.

Also included with WebSphere Personalization are new campaign management tools. Campaigns are sets of business rules that work together to accomplish a business objective.¹⁰⁰ Implicit profiling services can collect real time information about site visitor actions and then construct personalization business rules using this data. Implicit profiling tracks the areas of a site where a user is most active in order to determine his or her interests.

To analyze the effectiveness of the site and its personalization strategies, the server provides logs that can be analyzed by IBM Tivoli Web Site Analyzer, which can then create reports for the portal's business owner. This helps the company measure the effectiveness of business rules and campaigns in achieving its objectives.

¹⁰⁰ For example, an HR manager may want to run a campaign to encourage employees to enroll in a stock purchase plan.

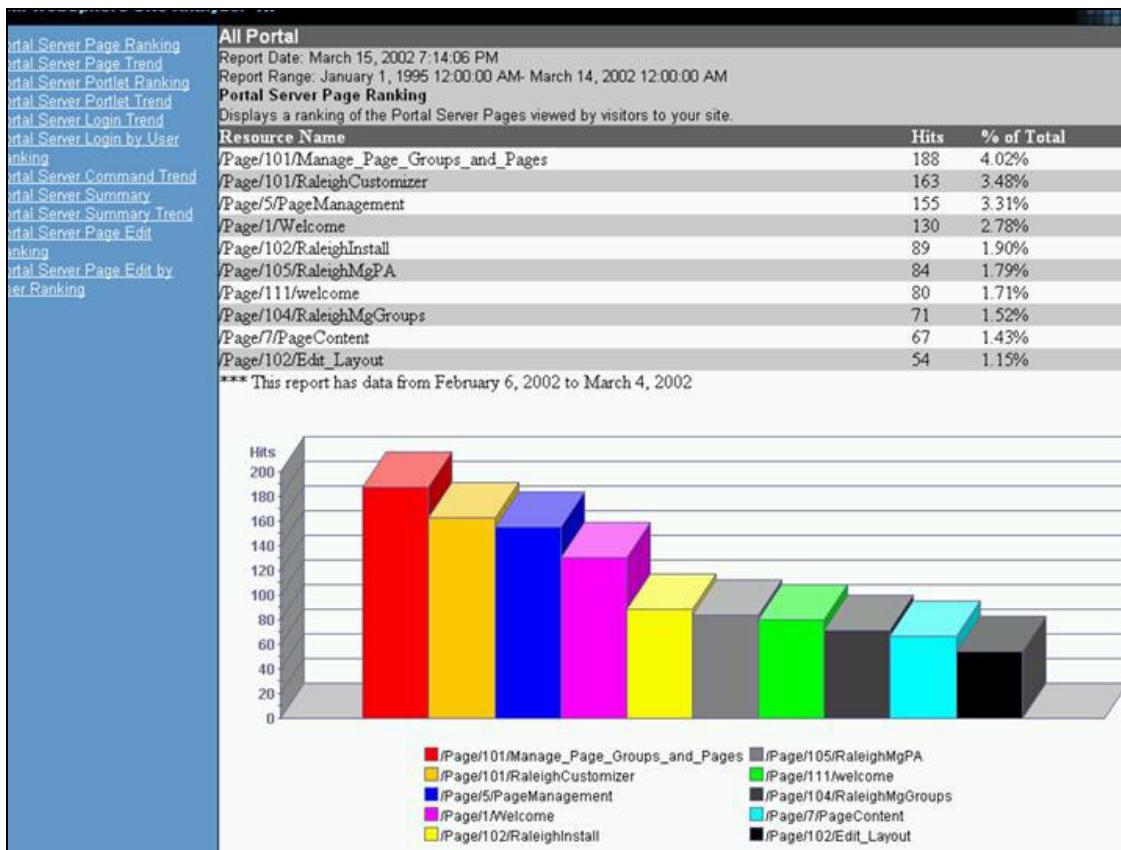


Figure B-2 A portal usage report from Web Site Analyzer

B.1.5 Administration

Administration of the portal is done through the portal itself in either a centralized or a delegated fashion. Administrators can deliver a new service to users simply by adding new portlets to the pages of the portal. Since these are portlets just like bookmarks or reminders or news or any other portlets, administrators can control access to them, place them on portal pages, and perform any of the usual steps.

Administrative portlets are provided for adding portlets to the portal's registry, managing users, groups, and access control lists; clipping Web pages; publishing Web services; setting portal-wide settings; managing logs; and performing other common tasks.

Portal settings

In the global settings portlet, administrators can change portlet settings such as the default language or cache time-out values. In this version of WebSphere Portal, there are some new settings that control how new user sessions are handled and what to do when a user tries to access a portlet without authorization. Unauthorized access can be ignored (in other words, the portlet is not displayed), or the portlet can be replaced by an informative message so that the user can take the necessary actions to correct the situation. Returning users might wish to pick up where they previously left off, so there is a setting to retain the state of the last visit and return to that page next time.

Web clipping portlet

One of the most important new portlets is the Web clipping portlet. This portlet is used to display sections of existing Web pages. You can visually select portions of the page or clip all the text between specific tags. This way, you can precisely control what markup is extracted. The portlet can optionally rewrite the links inside the clipped page, which is useful for displaying existing pages without leaving the portal's navigation structure. Each time you clip a Web page, a new portlet is created in the portal's registry. Whenever the new portlet is displayed, it retrieves the current version of the Web page and extracts the clipped portion to display.

Managing portlets

When installing new portlets you can use Web Archive (WAR) files from your local file system or install portlets that have been previously published in a UDDI directory.

Installing new portlets

The new portlet is automatically activated, but with no special permissions.

Managing the portlet catalog

Once the portlet is installed, you can copy it, set its configuration parameters, activate or deactivate it, or uninstall it.

Users and groups

In previous versions of WebSphere Portal, it was necessary to use your LDAP directory's administration tools to manage user and group information. Now, portlets have been added so that you can manage user and group information without leaving the portal. You can also manage a user's group memberships.

Web services

A *Web service* is an interface that describes a collection of network accessible operations. The interface is described using a standard XML description language called Web Service Description Language (WSDL) so that the service can be invoked without prior knowledge of the platform, language, or implementation design of the Web service. Web services are located using the standard Universal Description and Discovery Interface (UDDI), which may be a private or public registry of Web services. WebSphere Portal provides extensive support for Web services. Portal administrators can publish and bind remote portlets as Web services, making the remote portlets available in the portal's registry dynamically. When a remote portlet is used, its services are invoked using Simple Object Access Protocol (SOAP) or other transport protocols.

Federated portals

Consider a large corporation that has several different portals. Each of these portals offers portlets to its users and may choose to publish some of them as Web services, thus making them available for access through other portals. Completing the publishing step puts an entry for the portlet into a UDDI directory. An administrator at another portal can browse this directory to find all of the portlets that have previously been published and bind these into their local portal. This makes the portlet available as though it were locally installed. Of course, the portlet is actually running remotely on the original portal server that published it. The effect is to have a *federated* portal, where portlets may be running at any location in the network of portals.

Individual portlets can also use Web services internally to deliver their functionality. WebSphere Studio Application Developer provides development tools for quickly developing Web services and for generating proxy classes from WSDL descriptions.

Logging

Administrators can control the tracing and logging activity through the Log Files portlet and by modifying the configuration properties files of the logging subsystem. The portal server also records user activity in logs that can be processed by Tivoli Web Site Analyzer. Overall usage statistics such as logins and logouts are tracked, along with portlet and page usage statistics.

Table B-1 Portal Usage reports from Tivoli Web Site Analyzer

Project Summary	Displays summary statistics about the analysis, such as the total number of hits, sessions, visitors, and page views.
User Agent Ranking	Displays a ranking of user agents employed by visitors to your site.
Browser Ranking	Displays a ranking of browsers used by visitors to your site.
Platform Ranking	Displays a ranking of platforms employed by visitors to your site..
User Ranking	Displays a ranking of users to your site by frequency of visits.
Page Ranking	Displays a ranking of the pages viewed or edited by visitors to your site.
Portlet Ranking	Displays a ranking of the portlets viewed by visitors to your site.
Commands Ranking	Displays a ranking of the commands executed by visitors to your site, such as editing pages, creating new pages, or creating or deleting users and groups.
Logins	Displays the number of logins/logout into the portal.
Resource Ranking	Displays a ranking of the resources viewed by visitors to your site.

B.1.6 Collaboration

Corporate portals connect people to the applications, content, and resources they need. Portals also connect people to each other through community pages, shared bookmarks, contact lists, and personal productivity portlets.

Collaboration is really about people working in teams to solve a business problem. The portal server includes portlets and services designed to support the team and its activities with messaging, document libraries, user profiles, in-boxes and calendars, and electronic meetings. Users can access these collaborative services in the context of what they are currently doing rather than having to leave the portal to open another application.

For example, while working in the portal, users can easily see who else is online and then send an instant message, send an e-mail, or add a person to their contact lists. Collaboration portlets have advanced built-in features that allow portal users to take actions on documents or user names that appear in a portlet. Directly from the portlet, a portal user can see if other users are online and select from a menu of options to interact with another user.

Collaboration portlets

Collaboration portlets include Lotus Notes e-mail, calendar, and to-do list portlets, plus Lotus Notes discussion, document library, and TeamRoom portlets. There are also portlets for Discovery Server, Lotus Team Workplace, and Lotus Instant Messaging.

Several new portlets are being designed and prepared for an upcoming release on the portlet catalog. These include the People Finder, the Web Conference center, and the Team Workspaces portlet.

Table B-2 summarizes collaboration portlets.

Table B-2 Collaborative portlets

My iNotes™	Provides access to a Lotus iNotes server Welcome, Mail, Calendar, To Do List, Contacts, and Notebook functions.
My Notes Calendar, Mail, and To-do List	These portlets display the user's calendar, in-box, or to-do list from his or her mail database.
Notes Discussion	Views Lotus Notes databases built with the Discussion Database Template.
Notes View	Views can be configured to view most Lotus Notes databases.
Lotus QuickPlace	Displays a Lotus QuickPlace view inside the portlet.
Lotus Instant Messaging Chat	Displays a Lotus Instant Messaging chat window from within the portal.
Lotus Team Workplace	Displays a Lotus Team Workplace view inside the portlet.
Web Page	The Web Page portlet will show the contents of an Internet or intranet Web page in a scrollable portlet.
Discovery Server	Searches or displays a knowledge map.
People Finder	Locates a person by name or profile characteristics.
Team Workspaces	Show, search, and create Lotus QuickPlaces.
Web Conferences	Schedule, find, and attend e-meetings.

B.1.7 Related products

WebSphere Portal Extend integrates tightly with Lotus's world-class collaboration and knowledge management products—including Lotus Team Workplace, Instant Messaging, and Discovery Server—by providing portlets and services to access these products seamlessly from the portal.

Lotus Team Workplace

Lotus Team Workplace provides workspaces for sharing and organizing ideas, content, and tasks. It includes tools to manage team projects and schedules, organize discussion threads, and share documents.

The Lotus Team Workplace user interface is shown inside a portlet. The integration between the portal and the Lotus Team Workplace includes single sign-on so that users who access the portal can access the Lotus Team Workplace seamlessly, without requiring an additional login.

Lotus Instant Messaging

Lotus Instant Messaging provides instant messaging, shared whiteboards, and application sharing for electronic meetings. Lotus Instant Messaging functionality is integrated into the portal for access to chat sessions and buddy lists as well as for people and place awareness. Awareness is the ability to tell who the place members are and to find out whether they are online, offline, or not available.

Lotus Instant Messaging provides other services that can also be integrated through portlets: application sharing, whiteboarding, and online meetings.

Lotus Discovery Server

Lotus Discovery Server is a separately purchased product that creates expertise and knowledge maps by analyzing and categorizing documents. It creates profiles of users based on their document activity, including their topics of interest and their areas of expertise.

Discovery Server also examines user activity such as reading documents, responding to documents, timeliness of interactions, or links to specific documents. This way, Discovery Server can determine the relative proficiency of individuals to content categories. These proficiency indicators are called *affinities*, and they indicate the relative expertise of individuals to particular business areas of the organization.

Discovery Server continually assesses the strength of affinities using metrics, ensuring that individuals with recent and high-quality expertise ratings are found and presented to users seeking expertise when browsing the Discovery Server Knowledge Map interface.

Lotus Discovery Server is purchased separately from WebSphere Portal.

Collaborative Services

Lotus Collaborative Components are building blocks (APIs and JSP tag libraries) for integrating the functionality of Lotus Domino, Instant Messaging, Team Workplace, and Discovery Server into the portal. Developers can leverage their

features by using these components to add user interface extensions to portlets and portal pages.

The collaborative services hide the configuration details of the Lotus products installed within an enterprise. Instead of working with more complex product APIs, Lotus collaborative components provide developers with an easier method of integrating core collaborative features into any portal or portlet.

The benefit of using Collaborative Components is that they provide standardized access to Lotus applications, with easier APIs that are optimized for the portal. Security context is handled automatically, and upgrades of the back-end systems are transparent to the portlets. Collaborative Components work across versions of Domino 4.67 and above.

Table B-3 Lotus Collaborative Components

People Service	Provides online awareness capability so that portlets can determine who is online and what their status is.
Domino Data Service	Provides access to Domino data, including views, documents, and directory-related information. Use the services to locate Domino servers, create and edit documents, and work with views.
Discovery Services	Provides methods to find people, documents, knowledge maps, and profiles.
Lotus Team Workplace Service	Provides methods for creating a new Lotus Team Workplace
Menu Service	Displays a popup menu in the portal that is used with the People Service to create chat, e-mail, and search menus.

B.1.8 Application integration

By definition, a portal provides access to content, data, and services located throughout the enterprise. These include not only predefined connectors and portlets, but also tools for creating additional connectors and portlets.

Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) systems are excellent candidates for portlets because efficient, personalized access to these functions provides measurable return on your portal investment. WebSphere Portal includes portlets that help you access a variety of ERP and CRM systems.

Enterprise portlets

WebSphere Portal Application Integration provides a fast way to build new portlets that access various kinds of enterprise application systems, including

SAP, PeopleSoft, Siebel, and Oracle Financials. It also works for relational databases such as DB2, Oracle, Informix, and SQL Server.

The goals of the integrator framework were to provide a simple way for portlet administrators to build new portlets. Simply put, each tool is a portlet that builds new portlets for a specific back-end system. Operating the builder tool is simple and easy for administrators or business users. It works by querying the back-end system for metadata that describes the business object against which the new portlet will work.

Once the business object is selected, the person building the portlet selects which fields and operations he or she wants to enable in the portlet. These operations include searching, updating, deleting, and creating new records.

The screenshot shows the SAP Business Object Builder interface. At the top, there is a message about selecting fields for the portlet. Below it is a table titled "Select markup to configure" with columns: Selected, Display name, Field behaviour, Summary, Searchable, Type, and Namespace. The table lists various fields with checkboxes for selection and dropdown menus for behavior. At the bottom, there are buttons for Back, Next, and Cancel.

Selected	Display name	Field behaviour	Summary	Searchable	Type	Namespace
<input checked="" type="checkbox"/>	Appraisal Model Id	Appraisal Model Id	Read only	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	Appraisees	Appraisees	Allow update	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	Appraisers	Appraisers	Allow update	<input type="checkbox"/>		
<input checked="" type="checkbox"/>	Description	Description	Allow update	<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>	End-Date	End-Date	Allow update	<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>	Id	Id	Read only	<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>	Multi Source Text	Multi Source Text	Read only	<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>	Plan Version	Plan Version	Allow update	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/>	Start-Date	Start-Date	Allow update	<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>	Status	Status	Read only	<input checked="" type="checkbox"/>		

1-10 of 10

Back Next Cancel

Figure B-3 The SAP portlet builder

Once selections are made, the portal server stores the configuration information needed to run the portlet. No new Java code or service must be deployed—just the configuration information for each portlet.

Using this approach, anyone who understands the usage pattern of the enterprise application system can build a portlet in just a few minutes. In the

future it might be possible to build additional connectors for other applications such as Domino, Ariba, or Documentum.

Commerce

Many portals must include access to commerce functions such as portlets that access a catalog of products and services, or that present price, discount, or tariff information. Portals can provide these functions to consumers or to business partners for procurement or electronic exchange that links buyers and sellers of goods.

In January 2002, IBM announced a new capability for integrating WebSphere Portal and WebSphere Commerce Server to create commerce-enabled portals that provide consolidated, personalized access to commerce functions via the Web or wireless devices. This capability is available through service offerings from IBM and business partners.

The product roadmap for integrating commerce and portal functions includes integrating user management, authentication, single sign-on, and access control functions using common implementations and APIs for each. In the first step, WebSphere Portal and WebSphere Commerce Server will share a common user and group management subsystem. This way, users register only once so that all user profile information (for example, name, address, user ID, password, user preferences, etc.) are consistent across different portlets.

In addition, authentication for both the portal server and the commerce server will be based on WebSphere Application Server security, using either its built-in forms-based authentication or its Trust Association Interceptor modules to accept the authentication done by an external authentication proxy server. Once the users' credentials are established, the portal and commerce servers will use these for any access controls required by the different portlets.

B.1.9 Mobile portals

Next-generation portals will be accessible through more than just traditional desktop browsers. Access through handheld or mobile devices is becoming increasingly important. The portal server currently supports mobile devices by generating portal pages in three markup languages:

- ▶ HTML for desktop computers and some personal digital assistants
- ▶ WML for WAP devices, which are typically mobile phones
- ▶ cHTML for mobile devices in the NTT DoCoMo iMode network

Users can customize a unique home page for each device, selecting the content and applications most useful on the device. When a home page is requested, it is produced by first detecting the type of device that is making the request, and

then assembling the portlets (each of which render their contents in the appropriate markup language).

When a user customizes the home page for a particular device, the portlet selection list only shows portlets that can actually produce markup appropriate for that device. Thus, the list of available portlets for each device depends on what the portlets can actually do. Some portlets might be available for all supported devices, while others may be available only on a single device. The user interface design of each portlet varies from device to device, so that the user's experience can be fine-tuned. Thus, the user's home page and all portlets might be very different on a mobile phone versus a desktop browser.

Supporting new devices

The portal server supports several different markup languages so that portlets can render themselves for a variety of desktop and mobile browsers. Portlets that do not natively support the device markup can optionally be transformed using transcoding technology. This means that portlets can easily and automatically support mobile devices, even if the portlet developer did not explicitly support that device.

Clients and markups

Out of the box, the portal server's page aggregation subsystem supports several markup languages and recognizes certain browsers and mobile device user agent signatures. The framework for supporting markup languages is open and extensible, making it easy to support additional markups or new devices.

To support new browsers and devices, you add new markup and clients using the corresponding administration portlets. In the markups portlet, the markup name indicates the name of the folders used to store the page templates and theme or skin files matching that markup language.

B.2 Developer information roadmap

Note: The following is from the WebSphere Portal Zone, available to IBMers at <http://www7b.software.ibm.com/wsdd/zones/portal/roadmaps/develop.html>.

B.2.1 Basic skills

- ▶ Web development skills, including familiarity with HTML and XML
- ▶ Basic Java programming concepts
- ▶ Basic understanding of J2EE technology

J2EE technology and its component-based model simplifies enterprise development and deployment. The J2EE platform manages the infrastructure and supports the Web services required to enable the development of secure, robust, and interoperable business applications.

- ▶ Servlets and IBM WebSphere Application Server

Know how to install, configure, and maintain IBM WebSphere Application Server Advanced Edition and to deploy enterprise Java applications in a single machine or clustered configurations.

- ▶ WebSphere Application Server

Must be proficient in the foundation platform where WebSphere Portal runs.
The following is from

<http://www7b.software.ibm.com/wsdd/zones/was/bigpicture.html>:

WebSphere Application Server provides a rich, e-business application deployment environment with a complete set of application services, including capabilities for transaction management, security, clustering, performance, availability, connectivity, and scalability. WebSphere Application Server is a Java-based Web application server built on open standards that helps you deploy and manage applications ranging from simple Web sites to powerful e-business solutions. It is J2EE-compliant and provides a portable Web deployment platform for Java components, XML, and Web services that can interact with databases and provide dynamic Web content.

With the release of WebSphere Application Server Version 5, the IBM WebSphere Application Server family has evolved to a single, deployment-agile, Web services-enabled J2EE offering supported by complementary products such as an integrated development environment and an array of optional high-value platform solutions (such as portals, process management, and wireless). Whether you are just getting started or searching for a world-class application serving environment,

WebSphere Application Server Version 5 has an answer with one of its flexible configurations for your e-business needs.

B.2.2 Installation/configuration

Before you can create portlets, you must prepare your development environment by installing WebSphere Portal. You can perform a standard production installation of one of the WebSphere Portal for Multiplatforms offerings, or you can perform a customized development installation for use with the Portal Toolkit.

B.2.3 Migration

Just as migration from one version of WebSphere Portal to another can require special attention, there are also considerations you must take into account when migrating portlets and portlet applications for use with a different version of WebSphere Portal.

B.2.4 Developing portlets

To create a truly customized portal experience, you can create your own portlets and portlet applications.

Portlet API

Portlets are reusable components that provide access to Web-based content, applications, and other resources. Web pages, Web services, applications, and syndicated content feeds can be accessed through portlets. Companies can create their own portlets or select portlets from a catalog of third-party portlets. Portlets are intended to be assembled into a larger portal page, with multiple instances of the same portlet displaying different data for each user.

From a user's perspective, a portlet is a window on a portal site that provides a specific service or information (for example, a calendar or news feed). From an application development perspective, portlets are pluggable modules designed to run inside a portlet container of a portal server.

The *portlet container* provides a runtime environment in which portlets are instantiated, used, and finally destroyed. Portlets rely on the portal infrastructure to access user profile information, participate in window and action events, communicate with other portlets, access remote content, look up credentials, and store persistent data. The Portlet API provides standard interfaces for these functions. The portlet container is not a standalone container like the servlet container. Rather, it is implemented as a thin layer on top of the servlet container and reuses the functionality provided by that container.

IBM is working with other companies to standardize the Portlet API, making portlets interoperable between portal servers that implement the specification. The Portlet API offered in WebSphere Portal Version 4.1 is the first step toward the Portlet API standardization.

IBMer can go to

<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpspapi.html> for more information.

Portlet creation

IBMer can go to

<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpswrplt.html> to learn about the following:

- ▶ Creating a simple portlet instance
- ▶ Sample portlets
- ▶ Setting up a portlet development environment
- ▶ Compiling Java source
- ▶ Packaging and deploying portlets
- ▶ Generating markup
- ▶ Using persistence
- ▶ Portlet messaging
- ▶ Message and trace logging
- ▶ Refreshing the portlet cache
- ▶ Parallel portlet rendering
- ▶ Accessing the portlet session when the user is not authenticated
- ▶ Analyzing portlet usage
- ▶ Integrating advanced features with portlets

Guidelines for developing portlets

IBMer can go to

<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpspar.html> to learn about the following:

- ▶ Model-View-Controller (MVC) design pattern
- ▶ Portlet creation guidelines
- ▶ Markup guide
- ▶ Using Portlet API tags

Portlet security

IBMer can go to

<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpsadvdev.html> to learn about the following:

- ▶ Using the credential vault

- ▶ Credential vault samples
- ▶ Using JAAS to retrieve the user's credentials

Collaborative Components API

Lotus Software Collaborative Components provide Java API methods and tags for JavaServer Pages (JSPs) to extend the functionality of Lotus advanced collaboration in portlets. Collaborative Components are a set of methods and tags that allow developers who are writing portlets for WebSphere Portal Server or other application servers to add Lotus collaborative functionality to their portlets. Collaborative Components can be used to develop new custom portlets or to add collaborative functionality to existing portlets (for example, menus or people links indicating online status).

Collaborative Components are designed to provide access to the functionality of Lotus Software products enabled to work as companion products to WebSphere Portal Server. As such, the methods provided by Collaborative Services complement rather than replace the APIs of individual companion products such as the Domino Java API and the Discovery Server KDS API.

Collaborative Components provide standardized access to applications, easy-to-use APIs that are optimized for a collaborative portal, and a consistent security model across all Lotus Software companion products.

All of the Collaborative Components except for people and menu tags are UI-neutral. That is, portlet developers can design the user interface for the collaborative features they are implementing. The goal of Collaborative Components is to provide the data necessary for rendering the user interface and to allow the developer to execute actions on the Lotus collaborative products that have been installed and enabled in the portal environment.

The Java APIs in Collaborative Components contain no platform-specific code. Consequently, these components are UI-neutral and independent of implementation details. Because they are UI-neutral, Collaborative Components can be used to implement pervasive applications for mobile and wireless devices.

Since Collaborative Components hide the configuration details of the Lotus products installed within an enterprise, collaborative functionality can be easily added to a portlet in a generic way—regardless of the physical requirements and machine-dependent details of a portal configuration.

Application developers using Collaborative Components can design and implement functional extensions in portlets that incorporate the features of Lotus Domino and Notes, Team Workplace, Instant Messaging, and Discovery Server. Collaborative Components for other Lotus Software products will become

available in the future. JavaDoc for Collaborative Components is installed on the Portal Server. IBMers can go to http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/collab/ksd_cs_api_intro.html for more information.

Struts portlet framework

Struts is a Jakarta open source project that allows developers to efficiently implement their Web applications using a Model-View-Controller design pattern. Struts provides the Controller component, a collection of JavaBeans and helper classes for creating the Model, and a set of tag libraries for the JSP pages.

A Struts application is made up of actions, pages,¹⁰¹ and beans. The application writer creates these objects and defines in the *struts-config.xml* configuration file, the relationships between these objects, and the transitions that occur. These actions represent processing that occurs prior to going to another page. The configuration of an *ActionMapping* associates a path with an *Action* and with *ActionForms* and can list one or more destination pages following execution of the action. ActionForms are beans associated with actions, supplying the data to fill in various fields on a page. As a result of executing the action, an *ActionForward* object is returned containing the path to next request. Typically, the returned path is to a page, but it is also possible that it is to another Action.

The Struts Web site offers resources for developers, including a User Guide, informative JavaDoc, and the source code. IBMers can go to <http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpsstruts.html> for more information.

Page sequences

Page sequences is an extension to the Struts Portlet Framework that leverages much of the existing Struts functionality to support the design and implementation of multi-screen flows within a Struts application. This extension supports the generation of an event-driven screen-flow model that allows the developer to concentrate on solution-specific tasks.

Currently, Struts applications are left to manage navigation issues and deal with Forms whose logical contents cross page boundaries. Typical business transactions require more than one interaction with the user, therefore spanning more than one JSP page and action. Page sequences leverage much of the existing Struts functionality and extends the Struts framework to support the design and implementation of multi-screen flows within a Struts application, resulting in the following advantages:

- ▶ The configuration explicitly defines the exact flow of pages and actions.

¹⁰¹ Usually JSPs but sometimes HTML pages.

- The screen flow automatically ensures that the correct sequence is followed and that the user cannot back up and submit the same form multiple times.
- The screen flow supports separation of data from one screen flow to another.
- If the business logic is implemented properly, the order of screens within the flow can be changed without requiring major configuration and code changes.

IBMer can go to

<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpspageseq.html> for more information.

Click-to-Action portlets

Click-to-Action provides a framework for communication that simplifies users' interactions with portlets on a portal page. With a simple click, users can transfer data from a source portlet to one or more target portlets, causing the target to react to the action and display a new view with the results.

The Click-to-Action framework includes a runtime that automatically matches sources with compatible targets based on type information, and inserts clickable icons associated with sources on portlet pages. When users click on an icon next to a particular source, they are presented with a pop-up menu containing the list of targets for the action. After the user selects a specific target, the Click-to-Action runtime delivers the data to the target in the form of the corresponding portlet action. The portlet does not need to distinguish between an action initiated by user interaction with its own page segment and that initiated using the Click-to-Action route. This keeps the programming effort to a minimum, allowing Click-to-Action portlets to follow the normal portlet programming model.

JSP tag reference

The following tags are used by portlet or portal JSPs.

- Portlet API tags
- Portal JSP tags
- Struts WML tags
- Page sequence tags

IBMer can go to

<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wps.jspref.html> for more information.

More information

- *IBM WebSphere Portal V4 Developer's Handbook*, SG24-6897, found at <http://publib-b.boulder.ibm.com/Redbooks.nsf/9445fa5b416f6e32852569ae00>

[6bb65f/309f4c3c906a089c85256bf800630def?OpenDocument&Highlight=0, SG24-6897-00](http://publib-b.boulder.ibm.com/Redbooks.nsf/9445fa5b416f6e32852569ae00_6bb65f/309f4c3c906a089c85256bf800630def?OpenDocument&Highlight=0,SG24-6897-00)

This IBM Redbook helps you plan and develop portlet applications using the IBM WebSphere Portal Enable and Extend offerings. The information provided in this redbook targets B2E (Business-to-Employee) enterprise applications, but most of the scenarios presented apply to B2C applications as well. In this redbook you will find step-by-step examples and scenarios showing ways to integrate your enterprise applications into an IBM WebSphere Portal environment using the WebSphere Portal APIs provided by the Portal Toolkit to develop portlets. It also discusses extending your portlet capabilities to use other advanced functions such as themes and skins, personalization, search capabilities, content management, national language support, transcoding, and Web clipping.

Elements of the portlet API are described and sample code provided. The scenarios included in this redbook can be used to learn about portlet programming and as a basis for developing your own portlet applications. You will also find numerous scenarios describing recommended ways to develop portlets and portlet applications using the APIs provided by the IBM WebSphere Portal Toolkit.

A basic knowledge of Java technologies such as servlets, JavaBeans, EJBs, and JSPs, as well as XML applications and the terminology used in Web publishing, is assumed.

- ▶ A *Portal Composite Pattern Using WebSphere V4.1*, SG24-6869, found at [http://publib-b.boulder.ibm.com/Redbooks.nsf/9445fa5b416f6e32852569ae00_6bb65f/309f4c3c906a089c85256bf800630def?OpenDocument&Highlight=0, SG24-6869-00](http://publib-b.boulder.ibm.com/Redbooks.nsf/9445fa5b416f6e32852569ae00_6bb65f/309f4c3c906a089c85256bf800630def?OpenDocument&Highlight=0,SG24-6869-00)

The Patterns for e-business are a group of proven, reusable assets that can speed the process of developing applications. The Portal composite pattern combines Business and Integration patterns to help implement a portal solution. This IBM Redbook provides a technical scenario and guidelines for the Portal composite pattern. It shows how the Composite pattern works and documents the tasks required to build a technical scenario of it.

Part 1 of the redbook guides you through the process of choosing the Business and Integration patterns of the Composite pattern and then drilling down to the Application and Runtime pattern and Product mapping to deliver the desired functionality of the Portal composite pattern.

Part 2 provides a set of guidelines for building your portal application and includes a discussion of application design, application development, and systems management.

Part 3 demonstrates how to implement a portal solution via a technical scenario. This technical scenario uses the WebSphere Portal Extend offering.

► **Portal Toolkit**

Portals provide a mechanism for aggregating information and access to enterprise services into a single consolidated view for the Web. A portlet (similar to a servlet) provides access to a specific application or function being made available to the user via the portal.

The IBM Portal Toolkit Version 4.3 provides the capabilities to customize and manage the enterprise portal and to create, test, debug, and deploy individual portlets and Web content. Templates enable developers to quickly and easily create their own portlets. Debugging and deployment tools shorten the development cycle. Sample portlets that demonstrate best programming practices are also provided.

The Portal Toolkit plugs into the IBM WebSphere Studio Workbench, which provides a comprehensive framework for the development of e-business applications.

IBMers can go to

<http://www-3.ibm.com/software/info1/websphere/index.jsp?tab=products/portaltoolkit> for more information.

► **WebSphere portal catalog**

The WebSphere portlet catalog describes portlets created by numerous companies for use with WebSphere Portal. You can find portlets for your specific needs by searching or browsing by category.

IBMers can go to

<http://publib-b.boulder.ibm.com/Redbooks.nsf/9445fa5b416f6e32852569ae006bb65f/309f4c3c906a089c85256bf800630def?OpenDocument&Highlight=0,SG24-6869-00> for more information.

► **Portlet Development Guide: Second Edition**

This document shows how to develop a portlet using the Portlet API 1.2. It describes Portlet API concepts and elements, illustrating these concepts and API elements using examples ranging from a simple portlet with no output to a complex portlet application with more advanced features. Finally, it describes portlet development for the IBM WebSphere Portal Version 4.2 environment.

IBMers can go to

<ftp://207.25.253.53/1/wsdd/pdf/V42PortletDevelopmentGuide.pdf> to view/download.

B.3 What's new with Version 5.0

WebSphere Portal includes new functions and enhancements that make it easier to install, administer, design, and use the portal. Go to <http://pvcid.raleigh.ibm.com/wpf/ic/wpo500/ent/en/InfoCenter/index.html>.

B.3.1 New features

This section provides information about new features that are provided with WebSphere Portal.

- ▶ **New versions of software components to support the latest technology**

Many software components have been updated from the last release. Refer to the Components overview topic for details on the software versions included in this release.

- ▶ Additional database support

Database support has been extended to include additional database products. See the supported hardware and software topic to view the supported databases.

- ▶ **Document Manager to help you manage information**

Document Manager is a new document repository and management system that replaces Portal Content Organizer. Users navigate a hierarchy of folders that contain documents. Administrators can use Document Manager for project management tasks such as access control, document approval, and tracking versions of documents.

- ▶ **WebSphere Portal Productivity Components to help you author documents in the portal**

These components provide text editing, spreadsheet, and presentation functions. They allow authoring of documents directly within the portal. They are not intended to provide all the functionality associated with a full-fledged productivity application. Instead, they provide a version of the most widely used functionality typically available in a productivity application of this type.

- ▶ **Lotus Collaboration Center features that improve productivity**

Lotus Collaboration Center includes components that WebSphere Portal can use to find, connect, and work with people. Lotus Collaboration Center now includes the following:

- *People Finder portlet* provides an online directory for finding, contacting, and working with colleagues.
- *My Lotus Team Workplaces* includes support for Lotus Team Workplace.
- *Lotus Web Conferences* includes support for Lotus Instant Messaging.

- *Collaboration Center* now installs and deploys two portlets that were available separately in previous releases:
 - Lotus Instant Messaging Contact List
 - Lotus Instant Messaging Who Is Here
- A Collaboration Center sample page is now installed and deployed to provide a single page that displays all five Collaboration Center portlets.
- ▶ **New and enhanced business portlets to improve productivity**

The following portlets are included with WebSphere Portal:

 - *Internet Mail Box* (Mail Box) is new. This portlet allows the user to receive, compose, and send Internet e-mail using either a Web browser or wireless phone. The Mail Box allows the user to view supported files such as rich text documents, spreadsheets, and presentations. It allows loading and saving attachments to and from the Document Manager application. It also provides a Rich Text Editor to compose e-mails.
 - A single *Notes portlet* is now available. The Lotus Notes and Domino portlet represents a consolidation of multiple Notes portlets that were previously shipped.
 - A single *iNotes portlet* is now available. The Domino Web Access (iNotes) portlet represents a consolidation of multiple iNotes portlets that were previously shipped.
 - A *Newsgroup portlet* is now available for subscribing to and browsing Internet news groups.
 - The *MyList portlet* allows for simple check-off lists to be created and maintained.
- ▶ **New search features to improve search capabilities**

New search functions are included that can find, categorize, summarize, and support more than 200 document formats.
- ▶ **Portlet builders to help you create portlets**

Application portlet builders are included that enable users to create portlets that can access and manipulate data in an enterprise application.

B.3.2 Installation improvements

Improvements for the installation program include:

- ▶ **Changes to the installation program for quicker installation**

You can get WebSphere Portal up and running quickly. The installation program enables you to install the base portal without having to perform much planning or provide much input during installation.

In addition, the Cloudscape™ Java database is automatically installed for storing portal configuration data so that you can get up and running quickly. Because Cloudscape is automatically installed, you do not have to plan your database setup prior to installing WebSphere Portal. Of course, if you plan to use a different database, you can transfer your data after you install Portal.

► **New tasks to help you configure WebSphere Portal to use other software**

Configuration tasks are provided to get WebSphere Portal working in your specific environment. The configuration tasks are commands that you run to customize WebSphere Portal to use different software—such as collaborative components, a database other than Cloudscape, or an LDAP server for security. You can also use configuration tasks to alter the portal configuration without having to reinstall the product.

B.3.3 Improvements for administering and designing WebSphere Portal

This section provides information about improvements that can help you administer or design WebSphere Portal.

► **A redesigned administrative interface**

Because an administrative interface is shipped with WebSphere Portal, you do not have to design your own. The administrative interface has been redesigned to enhance your experience with the portal. Enhancements include:

- New administrative portlets
- Navigation improvements
- New themes and skins
- Updates to existing administrative and customizer themes and skins
- Context-sensitive links

► **Cooperative portlets**

Cooperative portlets is a new capability that enables portlet interaction on a page, including the Click-to-Action technology introduced in V4.2, but expanding on that feature to include more programming and user-interaction enhancements.

► **Enhanced support for managing user data**

WebSphere Portal uses Member Manager, a component that manages data for users and groups of users. Member Manager can access user information in different types of repositories such as supported LDAP directories and databases. Member Manager replaces the WebSphere Member Services that was shipped with previous releases of WebSphere Portal.

► **Improved function with XML access**

You can use XML access to configure the portal. XML access is the batch processing interface for portal configuration updates. You can use XML access to export entire portal configurations or parts of the configuration to an XML file,¹⁰² and to re-create the exported configurations from such a file. Because of the new use of Global Unique IDs (GUIDs) instead of Object IDs (OIDs) to identify portal resources, exported configurations are portable across WebSphere Portal installations. Function and documentation were both improved to help you better understand the use of XML access for portal configuration.

► **Support for user-friendly URLs**

WebSphere Portal now provides support for user-friendly URLs, which you can now define and map pages to using the portal administrative interface.

► **Improved handling of error messages**

WebSphere Portal has an improved logging mechanism based on WebSphere Application Server V5. Now error messages are localized to improve access and usability.

► **The ability to arrange content in a tree structure**

Portal administration is made more flexible by arranging portal pages in a tree hierarchy for administration. The new portlet access control component is sensitive to that tree structure so that permissions granted to a portal page are also respected by its child pages.

► **Portlet menus for improved navigation**

Portlet menus are an extension to the Portlet API that allow portlets to contribute menu items to the portal navigation. In this release of WebSphere Portal, helper classes have been provided that simplify the development of portlet menu items. These can be created using a static XML file or dynamically generated and updated with each request.

► **Transcoding**

WebSphere Portal allows administrators to adapt portal content for diverse situations through transcoding technology. Administrators can now use transcoding at the portlet level.

► **Portal Toolkit**

Portal Toolkit includes updates for WebSphere Version 5 as well as updated portlet wizards. New portlet examples are also included.

¹⁰² For example, specific pages

- ▶ **Web Clipping Portlet updates**

Programmaing changes improve the performance of the Web Clipping portlet and simplify the creation of other clipping portlets.

B.3.4 Removed features

Support for portlets published as remote Web services has been removed for this release. This support is expected to be provided in the future. Please check release notes for updates.

B.3.5 Contents

This section provides information about the products and components included with WebSphere Portal.

WebSphere Portal

WebSphere Portal is a J2EE application that runs on WebSphere Application Server. Its main function is to serve the WebSphere Portal framework to the desktops and mobile devices of portal users. WebSphere Portal creates an environment that provides the connectivity, administration, and presentation services required. WebSphere Portal includes several new functions and enhancements that make it easier to design, administer, and use.

WebSphere Portal Content Publishing

WebSphere Portal Content Publishing provides a Web content management solution that gives nontechnical users greater control over content published to portals and Web sites. Users benefit from the combined power of having one place to manage content for their Portal environment or other Web sites and an easy-to-use Web interface. This interface puts content management back into the hands of nontechnical business users and provides them with tools (such as personalization rules, templates, workflow, and versioning) that make the content creation process simple yet controlled. WebSphere Portal Content Publishing decreases Web maintenance and administration costs, increases sales and profits by deploying timely and personalized content, and improves efficiency by getting all content produced in an enterprise to the Web.

WebSphere Portal Content Publishing includes a variety of functions. The core part of WebSphere Portal Content Publishing involves the content creation process. The Web browser interface enables users to contribute content to a Web site with little training in a simple and controlled manner. This solution allows large teams to work together, jointly where necessary, but with enough isolation to manage content changes in their respective work groups until they are ready to share them with the extended contribution team. Versions of

changed content can be stored in the versioning repository for later reuse. WebSphere Portal Content Publishing manages the task lists of each user, maintains access control over what content each user can see or change, and coordinates the approval and publishing process when content has been approved. This is all accomplished through a workflow process that can be customized to meet the needs of each individual customer.

WebSphere Portal Content Publishing includes two personalization technologies to dynamically tailor Web content to meet the needs of different users and groups:

- ▶ Rules-based filtering to display selected content to targeted users
- ▶ Matching techniques that extract visitor behavior and tailor the content within individual portlets based on that behavior

This technology creates a customized user experience for improved employee productivity and more effective consumer messaging. To augment the personalization features, WebSphere Portal Content Publishing provides a reporting mechanism that assesses the effectiveness of business rules and builds reports for the WebSphere Portal Content Publishing user to leverage when analyzing the impact of targeted content on consumers. The reporting mechanism provides data to Web site managers to help them revise targeting strategies based on user response to content they have received.

Lotus Collaborative Components

Lotus Collaborative Components are UI-neutral API methods and tag libraries that allow developers who are writing portlets for WebSphere Portal to add Lotus Software collaborative functionality to their portlets. Application developers using Collaborative Components can design and implement UI extensions that leverage the features of Domino, Lotus Team Workplace, Lotus Instant Messaging, and Lotus Discovery Server.

Portal Toolkit

Portal Toolkit provides the capabilities to customize and manage the enterprise portal and create, test, debug, and deploy individual portlets and Web content. Templates enable developers to quickly and easily create their own portlets. Debugging and deployment tools shorten the development cycle. Sample portlets that demonstrate best programming practices are also provided.

Portal Toolkit plugs into WebSphere Studio Site Developer or WebSphere Studio Application Developer, providing a comprehensive framework for the development of e-business applications.

A copy of WebSphere Studio Site Developer licensed for use only with Portal Toolkit is included. Consult the product license for details.

IBM WebSphere Application Server Version 5.0 Fix Pack 1

IBM WebSphere Application Server is a Web application server that provides J2EE services for the WebSphere Portal environment. It executes the Java portlets, JavaBeans, JSP files, and Enterprise Java Beans (EJBs) used by WebSphere Portal. This component is the platform upon which WebSphere Portal runs.

IBM Directory Server Version 5.1

IBM Directory Server is a Lightweight Directory Access Protocol (LDAP) directory that runs as a standalone daemon. In the WebSphere Portal environment it stores, updates, and retrieves user-specific data related to authentication, such as user IDs and passwords.

DB2 Enterprise Server Edition Version 8.1 Fix Pack 1

DB2 Enterprise Edition is a Web-enabled relational-database management system. In the WebSphere Portal environment, DB2 stores portal configuration data as well as portal-specific, access-control, and user data.

IBM WebSphere Studio Site Developer Version 5.0.1 Fix Pack 1

IBM WebSphere Studio Site Developer is an Integrated Development Environment (IDE) for building, testing, and deploying J2EE applications. It provides integrated development support for building J2EE applications with HTML pages, servlets, JSPs, and EJBs. It also creates Web service applications with open standards, generates XML documents from DTDs, and enables a collaborative team development environment. It enables end-to-end local and remote testing and creates high-quality applications using wizards, code generators, and best practices.

WebSphere Translation Server Version 5.0

Translation Server is an MT (Machine Translation) offering that helps you remove language as a barrier to global communication and e-commerce. It enables enterprises to provide Web pages, e-mail messages, and chat conversations in multiple languages and in real time. Specifically designed for enterprise use, the Translation Server allows companies to leverage their existing Web infrastructure to provide content to users in their native language at a fraction of the cost of professional translation.

Based on IBM machine translation technology, Translation Server is designed for scalability on multiple platforms. It can run on a dedicated server using RMI and Java protocol to communicate with the WebSphere Application Server. Web page HTML content can also be translation-enabled to support HTTP servers from Netscape, Apache, Microsoft, or IBM.

Translation Server consists of:

- ▶ MT engines for translating text from one language into another language (such as English to Spanish)
- ▶ User Dictionary Manager tools that allow specific words to be added to a domain (for example, tuning for a specific application or adding of slang or domain-specific phrases and terms)
- ▶ Support for WebSphere Application Server and HTTP servers from Microsoft, Netscape, Apache, and IBM

B.3.6 Available with WebSphere Portal Extend

The following products and components are included with WebSphere Portal Extend:

Lotus Collaboration Center Version 5.0

WebSphere Portal featuring Collaboration Center capabilities offers an integrated framework of e-workplace components for finding, connecting, and working with people:

- ▶ People Finder portlet
- ▶ Directory Connector
- ▶ Sample configuration
- ▶ My Lotus Team Workplaces portlet
- ▶ Lotus Web Conferencing (Lotus Instant Messaging) portlet

Using these components, users have immediate access to a searchable directory of people that is integrated with their workplaces and their e-meetings within the collaborative portal. Users can find people in the directory, see their online status, and interact with them using instant messaging and other actions provided by people links. In addition to search features, the People Finder provides views of each person's directory record and his or her place in the organizational context. People links are visible in workplaces and e-meetings to make employee interaction fast and easy, improving personal and organizational productivity.

IBM Lotus Domino Enterprise Server 5.0.12

IBM Lotus Domino Enterprise Server and Notes is groupware software that provides messaging and collaboration features. Lotus Notes is the e-mail, calendaring, group scheduling, Web access, and information management client. Domino is the integrated messaging and Web application server.

Domino Directory can provide your primary corporate directory or the secondary LDAP service that complements another LDAP service ¹⁰³ in your portal

environment. As a secondary LDAP service, Domino Directory provides user access to Lotus Notes-based information sources for Collaborative Components and collaborative portlets.

Usage restrictions apply. You are authorized to install and use Domino solely and exclusively in connection with your use of Lotus Instant Messaging and Lotus Team Workplace. Consult the product license for details.

Instant messaging and online awareness

Instant messaging and online awareness (also known as *people awareness*) is available. Using Lotus Instant Messaging, portal users can discover if others are available to chat, receive e-mail, share applications and other tools in an e-meeting or Web conference, and (if Lotus Discovery Server is enabled) find documents authored by others and display expertise profiles from the Knowledge Map.

Usage restrictions apply. For example, the e-meeting capabilities of Lotus Instant Messaging are not licensed for use. Consult the product license for details.

Virtual teamrooms

Note: Virtual teamrooms are based on Lotus Team Workplace.

A Web-based solution is available for creating team workspaces for collaboration. Using Lotus Team Workplace, portal users can work securely with colleagues, suppliers, partners, and customers. Lotus Team Workplace provides teams with workspaces where they can reach consensus through discussions, collaborate on documents, and coordinate plans, tasks, and resources.

IBM Tivoli Web Site Analyzer Version 4.5

IBM Tivoli Web Site Analyzer is a Web application that captures and analyzes Web site data to provide useful reports about visitor traffic, visitor behavior, site usage, site content, and site structure. You can build customized reports from predefined report elements or build custom reports that collect information specific to your site. Installed as a Web application on WebSphere Application Server, Tivoli Web Site Analyzer allows you to create trend reports that show Web site visitor traffic, site usage growth, and changes over time.

The new enhancements in Tivoli Web Site Analyzer include integrated support for WebSphere Portal. This support includes specific report elements that enable you to analyze portal usage data such as a ranking of the portal pages viewed by visitors and portal login trends. Using this integration involves configuring the

¹⁰³ For example, IBM Directory Server, Windows 2000 Active Directory, or Sun ONE

portal to generate application level logs, importing the logs to Tivoli Web Site Analyzer, and creating reports.

IBM Lotus Extended Search 4.0

IBM Lotus Extended Search is a Web-based solution that supports searching across distributed, heterogeneous, structured, and unstructured data sources through a single point of access. Although it does not require you to create a central index for the sources that you want to search, Lotus Extended Search serves as a complimentary solution to products that provide indexing services such as Lotus Discovery Server.

By entering a single request from a Web browser, you can search potentially thousands of data repositories and the Internet, simultaneously locating people with expert knowledge. Lotus Extended Search handles the logistics required to access many diverse sources simultaneously without requiring that you know any details about the various search interfaces. It also produces a single, ranked result set that contains responses from multiple sources. Links in the result set enable you to go directly to documents of interest or download attachments.

Lotus Extended Search can search and retrieve documents from the following types of sources:

- ▶ Eighteen popular Web search and news sites
- ▶ Lotus repositories such as Notes databases, Domino.Doc libraries, Lotus Team Workplaces, and Lotus Discovery Server knowledge maps
- ▶ Mail systems such as those managed by Lotus Notes and Microsoft Exchange
- ▶ Instant messaging systems such as Lotus Instant Messaging
- ▶ ODBC-compliant relational databases such as DB2, Oracle, Sybase, and MS SQL Server
- ▶ Document management systems such as IBM DB2 Information Integrator for Content¹⁰⁴
- ▶ File systems
- ▶ LDAP-compliant directories
- ▶ Full text indexes such as those created with WebSphere Portal, Domino Domain Index, Microsoft Index Server, and Microsoft Site Server

With the Lotus Extended Search C++ and Java API you can extend support to other types of sources that you need to search—such as proprietary, commercial, or legacy databases in your enterprise not mentioned in the preceding list.

¹⁰⁴ Enterprise Information Portal

B.3.7 Supplementary products

Though the following products are not included with WebSphere Portal, they can further enhance the features and functions provided:

IBM Tivoli Access Manager for e-business Version 4.1

IBM Tivoli Access Manager for e-business is an authorization and management solution that scales across the entire enterprise. A robust and secure policy management tool for e-business and distributed applications, it addresses the challenges of escalating security costs, growing complexity, and the need for uniform security policies across platforms. Tivoli Access Manager unites core security technologies around common security policies to help reduce implementation time and management complexity, thereby lowering the total cost of security-enhanced computing.

IBM Content Manager Version 8.2

IBM Content Manager is an unstructured content management system optimized for large collections of large objects—from scanned images, facsimiles, and PC files to XML and rich multimedia and Web content. IBM Content Manager also provides folder management and document workflow and provides the content infrastructure for applications—from call centers, high-volume claims processing, and accounts payable to e-commerce catalogues and e-learning.

Lotus Discovery Server

Lotus Discovery Server is a comprehensive knowledge server that lets portal users search or browse for information and subject matter experts (SMEs) from multiple locations, collaborate with colleagues instantly, increase knowledge-sharing, and decrease time spent looking for needed resources—all from a Web browser.

Lotus Discovery Server finds content, discovers relationships, locates expertise, and returns more accurate results through automated collaborative filtering based on usage. It does this on a backdrop of collaborative tools that instantly link people through awareness and instant messaging, allowing you to:

- ▶ Search and find answers quickly while avoiding information overload
- ▶ Identify documents, people, and places with a single search across your entire organization
- ▶ Capture and quickly catalog information and intellectual assets to avoid duplicate work
- ▶ Identify the experts for collaboration—on demand—to improve decision cycle times and to leverage skills across departmental and geographical boundaries.

WebSphere Edge Server

The WebSphere Edge Server architecture is an extension to the WebSphere software programming model that brings the WebSphere software platform to the edge of the network. It enables unprecedented performance, scalability, and availability for sophisticated e-business applications and includes the following features:

- ▶ Application offload
- ▶ Content distribution
- ▶ Enhanced caching
- ▶ Enhanced load balancing
- ▶ Security
- ▶ Transactional quality of service

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see "How to get IBM Redbooks" on page 434. Note that some of the documents referenced here may be available in softcopy only.

Collaboration-specific redbooks

- ▶ *WebSphere Portal 4.12 Collaboration Services*, [REDP0319](#) - A good overview of the collaborative capabilities provided out of the box with portal
- ▶ *Patterns: Custom Designs for Domino & WebSphere Integration*, [SG24-6903-00](#) - A good overview of best practice integrations of Websphere and Domino.
- ▶ *Working with the Lotus Instant Messaging Community Server Toolkit*, [SG24-6667-00](#) - The introductory chapters have a good overview of the Lotus Instant Messaging server architecture.
- ▶ *Portalizing Domino Applications* (coming soon) - Will cover in depth moving Domino apps to Portal. First 1-2 chapters will provide a good architectural overview of the various approaches to making such a "migration".

General portal redbooks

- ▶ *A Portal Composite Pattern Using WebSphere V4.1*, [SG24-6869-00](#) - Provides a good understanding of the business requirements solved by a portal via patterns.
- ▶ *IBM WebSphere Portal V4.1 Handbook Volume 1*, [SG24-6883-00](#) - The first several chapters provide a basic architectural overview of portal.
- ▶ *IBM WebSphere Portal V4 Developer's Handbook*, [SG24-6897-00](#) - The first several chapters provide a basic architectural overview of portal.
- ▶ *WebSphere Commerce Portal V5.4 Solutions Guide, Integrating WebSphere Commerce V5.4, Business Edition and WebSphere Portal V4.2*, [SG24-6890-00](#) - Good example of B2C portal solutions.

- ▶ *Edgeable Portal Industry Solution*, [ZG24-6723-00](#) - A good example of a real world WebSphere Portal Server V4 based solution. Available internally only.

Redbook Hints & Tips

- ▶ “How to Portalize Your Existing Web Applications,” [TIPS0028](#)

Other publications

These publications are also relevant as further information sources:

- ▶ *Patterns for e-business*, IBM Press, August 2001 (ISBN 1-931182-02-7)
- ▶ *Enterprise Java Programming with IBM WebSphere*, Addison-Wesley, October 2003 (ISBN 0-321185-79-X)
- ▶ *An Introduction to Web Application Development with IBM WebSphere Studio*, MC Press, October 2003 (ISBN 1-931182-11-6)
- ▶ *Integrated Solutions with DB2*, Addison-Wesley, November 2003 (ISBN 0-201754-85-1)
- ▶ *Building Applications with IBM WebSphere Studio and JavaBeans*, IBM Press, May 2003 (ISBN 1-931182-14-0)
- ▶ *Developing J2EE Applications with IBM WebSphere Studio*, IBM Press, October 2003 (ISBN 1-931182-10-8)

Online resources

These Web sites and URLs available to IBMers are also relevant as further information sources:

- ▶ “Administration Made Easier—Scheduling and Automation in DB2 Universal Database”
<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www7bsoftware.ibm.com/dmdd/library/techarticle/0301gartner/0301gartner.html&origin=ondemand>
- ▶ “Autonomic computing—Creating self-managing computing systems”
<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www-3.ibm.com/autonomic/index.shtml&origin=ondemand>
- ▶ “Autonomic computing—The value of self managing systems”
<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www-3.ibm.com/software/tivoli/features/oct2002/autonomic.html&origin=ondemand>
- ▶ Autonomic Computing Overview and Resources (Tivoli Developer Domain)

http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www-106.ibm.com/developerworks/tivoli/autonomic/library/1016/1016_autonomic.html&origin=ondemand

- ▶ Business Portals—WebSphere Portal

<http://www.ibm.com/websphere/portal>

http://www-3.ibm.com/software/info1/websphere/index.jsp?tab=products/portal&S_TACT=102BBW01&S_CMP=campaign

- ▶ Dassault Systèmes PLM Solutions—CATIA

http://www.3ds.com/en/brands/catia_ipf.asp

IBM site

<http://www-3.ibm.com/software/applications/plm/catiav5/>

- ▶ Dassault Systèmes PLM Solutions—ENOVIA Products

<http://plm.3ds.com/en/enovia/products/overview.asp>

IBM site

<http://www-3.ibm.com/software/applications/plm/enovia/>

- ▶ “Developing Grid computing applications, Part 1—Introduction of a Grid architecture and toolkit for building Grid solutions”

<http://www-106.ibm.com/developerworks/library/ws-grid1/>

- ▶ e-business Advisor on demand Offerings

<http://w3.ncs.ibm.com/sitelet.nsf/ondemand/JSHW-5MZUEN?OpenDocument&page=MP>

- ▶ e-business on demand—A developer's roadmap

<http://www-106.ibm.com/developerworks/ibm/library/i-ebodov>

- ▶ “From Data Management to Information Integration: A Natural Evolution” (IBM developerWorks)

<http://www7b.software.ibm.com/dmdd/library/techarticle/0206roth/0206roth.html>

- ▶ Galaxia ePCM

www.galaxia.ca/epcm

- ▶ “How Tivoli software products support the IBM Autonomic Computing Initiative”

[ftp://ftp.software.ibm.com/software/tivoli/whitepapers/wp-autonomic.pdf](http://ftp.software.ibm.com/software/tivoli/whitepapers/wp-autonomic.pdf)

- ▶ “How to decide if portlets are your best option” (IBM developerWorks)

<http://www-106.ibm.com/developerworks/ibm/library/i-portletintro/>

- ▶ “IBM Federated Database Technology” (IBM developerWorks)

<http://www7b.software.ibm.com/dmdd/library/techarticle/0203haas/0203haas.html>

- ▶ IBM Global Services Business Consulting Services
<http://w3.ibm.com/services/bcs/>
- ▶ IBM Grid Toolbox
<http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www.alphaworks.ibm.com/tech/gridtoolbox&origin=ondemand>
- ▶ IBM Patterns for e-business (IBM developerWorks)
<http://www.ibm.com/developerWorks/patterns/>
- ▶ IBM Patterns for e-business—Business patterns for Simple Implementations
<http://www-106.ibm.com/developerworks/patterns/select-pattern.html>
- ▶ IBM Patterns for e-business—Review Runtime patterns
<http://www-106.ibm.com/developerworks/patterns/portal/access-sso-runtime.html>
- ▶ IBM Software Support
<http://www.ibm.com/software/support>
- ▶ “Installing and configuring the IBM Grid Toolbox” (tutorial)
http://www-106.ibm.com/developerworks/cgi-bin/click.cgi?url=http://www-1.ibm.com/servers/esdd/tutorials/grid_computing.html&origin=ondemand
- ▶ “LDD Today—A Preview of Lotus Discovery Server 2.0” (IBM developerWorks)
<http://www-10.lotus.com/ldd/today.nsf/lookup/DSpreview>
- ▶ Portlet Development Guide—Second Edition
<http://wilson.boulder.ibm.com/wsdd/zones/portal/portlet/portletdevelopmentguide.html>
- ▶ Rational Software
<http://www.rational.com/>
- ▶ Resource center—e-business on demand (IBM developerWorks)
<http://www-106.ibm.com/developerworks/ondemand/>
- ▶ “Understanding the foundation of JSR-109” (IBM developerWorks)
<http://www-106.ibm.com/developerworks/webservices/library/ws-jsrart/>
- ▶ “Understanding Web Services” (IBM developerWorks)
http://www7b.software.ibm.com/wsdd/library/techarticles/0307_ryman/ryman.html
- ▶ WebSphere Application Server—The Big Picture (IBM developerWorks)
<http://www7b.software.ibm.com/wsdd/zones/was/bigpicture.html>
- ▶ WebSphere High-Volume Web Site (IBM developerWorks)

- <http://www7b.software.ibm.com/wsdd/zones/hvws/>
<http://www.ibm.com/websphere/developer/zones/hvws/>
- ▶ “WebSphere Portal—Collaborative Components API”
http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/collab/ksd_cs_api_intro.html
- ▶ “WebSphere Portal—Guidelines for developing portlets”
<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpspar.html>
- ▶ “WebSphere Portal—JSP tag reference”
<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpsjspref.html>
- ▶ “WebSphere Portal—Page sequences”
<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpspageseq.html>
- ▶ “WebSphere Portal—Portlet API”
<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpspapi.html>
- ▶ “WebSphere Portal—Portlet security”
<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpsadvdev.html>
- ▶ “WebSphere Portal—Struts Portlet Framework”
<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpsstruts.html>
- ▶ “WebSphere Portal—Writing portlets”
<http://publib.boulder.ibm.com/pvc/wp/42/exp/en/InfoCenter/wps/wpswrplt.html>
- ▶ WebSphere Portal Catalog
https://www-3.ibm.com/services/cwi/portal/_pagr/105/
- ▶ WebSphere Portal Enable for Multiplatforms
<http://www-3.ibm.com/software/genservers/portal/enable/>
- ▶ WebSphere Portal Experience Solution
http://tuscarora.raleigh.ibm.com/wps2.1/InfoCenter/wpf/high_welcome.html
- ▶ WebSphere Portal Express for Multiplatforms
<http://www-3.ibm.com/software/genservers/portalexpress/>
- ▶ WebSphere Portal Extend for Multiplatforms
<http://www-3.ibm.com/software/genservers/portal/extend/>
- ▶ WebSphere Portal InfoCenter
<http://submit.boulder.ibm.com/wsdd/zones/portal/links.html>

- ▶ WebSphere Portal Server documentation—Portlet development
http://tuscarora.raleigh.ibm.com/port_dev.html
- ▶ WebSphere Portal Toolkit
http://www-3.ibm.com/software/info1/websphere/index.jsp?tab=products/portal_toolkit
- ▶ WebSphere Portal Zone (WebSphere Developer Domain)
<http://submit.boulder.ibm.com/wsdd/zones/portal/>
<http://www7b.software.ibm.com/wsdd/zones/portal/roadmaps/develop.html>
- ▶ “What you need to know now about autonomic computing, Part 1—An introduction and overview” (IBM developerWorks)
<http://www-106.ibm.com/developerworks/ibm/library/i-autonom1/>

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Abbreviations and acronyms

ABLE	Agent Based Learning Environment	cHTML	Compact HTML for Small Information Appliances
ACL	Access Control List	CICS	Customer Information Control System
ADA	American Disability Act	CIS	Customer Information Service
ANSI	American National Standards Institute	CODASYL	Conference on Data Systems Languages
AOD	Architecture Overview Diagram	CORBA	Common Object Request Broker Architecture
API	Application Program Interface	COTS	commercial off-the-shelf
ASP	Active Server Page	CRM	Customer Relationship Management
AST	Automatic Summary Table	CSV	comma-separated values
B2B	Business-to-Business	DBMS	database management system
B2C	Business-to-Consumer	DII	Dynamic Invocation Interface
B2E	Business-to-Employee	DMU	Digital Mock-Up
BCS	IBM Global Services Business Consulting Services	DMZ	<i>demilitarized zone</i> (A computer host or small network inserted as a “neutral zone” between a company’s private network and the outside public network. It prevents outside users from getting direct access to a server that has company data. A DMZ is an optional and more secure approach to a firewall and effectively acts as a proxy server as well.)
BEA	BEA Systems. The acronym BEA is composed of the first initial of each of the company’s three founders: Bill Coleman, Ed Scott, and Alfred Chuang.		
BI	Business Intelligence		
BOL	Bill of Lading		
BOM	Bill of Materials		
BPA	Business Process Automation		
BPEL4WS	Business Process Execution Language for Web services		
BPM	Business Process Management	DSML	Directory Services Markup Language
CAD	Computer Aided Design	DTD	Document Type Definition
CAM	Computer Aided Manufacturing	EAI	Enterprise Application Integration
CGI	Common Gateway Interface	EAR	Enterprise Application Archive
		ebXML	Electronic Business XML

ECO	Engineering Change Order	IIOP	Internet Inter-ORB Protocol
ECR	Engineering Change Request	IML	Initial Machine Load
EDI	Electronic Data Interchange	IMSTM	Information Management System Transaction Manager
EDS	Electronic Data Systems Corporation	IPL	Initial Program Load
EIP	Enterprise Information Portal	IRC	Internet Relay Chat
EJB	Enterprise JavaBean	ISV	independent software vendor
ePCM	Galaxia Enterprise-wide Process Change Management	IT	Information Technology
ERM	Enterprise Resource Management	ITIL	IT Infrastructure Library
ERP	Enterprise-wide Resource Planning	ITSO	International Technical Support Organization
ES	Lotus Extended Search	IVP	Industry Value Project
ESRI	Environmental Systems Research Institute	J2EE	Java 2 Enterprise Edition
ETTK	Emerging Technologies Toolkit	JAAS	Java Authentication and Authorization Service
GIS	Geographic Information System	JAX-RPC	Java API for XML-Based Remote Procedure Calls
GUID	Global Unique ID	JCA	Java Connector Architecture
HP-UX	The UNIX-based operating system for the HP 9000 series of business servers from Hewlett-Packard	JCP	Java Community Process
		JDBC	Java Database Connectivity
		JMS	Java Message Service
		JMX	Java Management Extensions
		JNDI	Java Naming and Directory Interface
HR	Human Resources	JSP	JavaServer Page
HTML	Hypertext Markup Language	JSR	Java Specification Request
HTTP	Hypertext Transfer Protocol	JXTA	Pronounced “juxta,” this is Sun Microsystem’s peer-to-peer initiative, a research effort created to promote and explore new approaches to distributed computing. The organization’s name was developed as a short form of the word “juxtapose,” because to <i>juxtapose</i> is to put things next to each other, which is what peer-to-peer is all about.
HTTPS	HTTP over Secure Sockets Layer		
HVWS	High Volume Web Site		
IDE	Integrated Development Environment		
IDI	IBM Directory Integrator		
IDL	Interface Definition Language		
IETF	Internet Engineering Task Force		
IFB	International Food and Beverage		

K-map	Knowledge Map	PDM	Product Data Management
LCA	ENO VIA	PKI	Public Key Infrastructure
LDAP	Lightweight Directory Access Protocol	PLM	Product Life Cycle Management
LOB	Line of Business	POS	Point-of-Sale
LTPA	Lightweight Third Party Authentication	PPR	ENO VIA
MAPE	monitor, analyze, plan, and execute	QAD	IBM Business Partner
MED	Management of External Data	QBIC	Query by Image Content
MIME	Multi-Purpose Internet Mail Extensions	RFC	Request for Change
MS	Microsoft	RIXML	Research Information Exchange Markup Language
MT	Machine Translation	RMI	Remote Method Invocation
MVC	Model-View-Controller	ROI	Return on Investment
NASSL	Network Accessible Service Specification Language	RPC	Remote Procedure Call
NFRs	Non-Functional Requirements	RSS	Rich Site Summary
NTT	Nippon Telegraph and Telephone	RUP	Rational Unified Process
OASIS	Organization for the Advancement of Structured Information Standards	SAM	Strategic Alignment Model
OCS	Open Content Syndication	SCL	SOAP Contract Language
ODBC	Open Database Connectivity	SCM	Supply Chain Management
ODS	Operational Data Store	SDL	Service Description Language
ODW	On Demand Workplace	SEI	Service Endpoint Interface
OGSA	Open Grid Services Architecture	SGML	Standard Generalized Markup Language
OGSI	Open Grid Services Infrastructure	SLA	Service Level Agreement
OID	Object ID	SMB	Small and Medium Business
OLAP	Online Analytical Processing	SME	Subject Matter Expert
OLTP	Online Transaction Processing	SMS	Short Message Service
PARA	Parallel Query Processing Techniques	SMTP	Simple Mail Transport Protocol
PCM	Product Change Management	SNMP	Simple Network Management Protocol
PDA	Personal Digital Assistant	SOAP	Simple Object Access Protocol
		SQL	Structured Query Language
		SQLMM	SQL Multimedia
		SQLX	XML extensions to SQL
		SSM	Signature Selling Method

SSO	Single Sign-On	XSD	XML schema
SSR	Signature Sales Resource	XSL	Extensible Stylesheet Language
TAI	Trust Association Interceptor		
TAM	Tivoli Access Manager		
TAM	Tivoli Access Manager		
TCO	Total Cost of Ownership		
TCP/IP	Transmission Control Protocol/Internet Protocol		
TeAM	Technical e-business Architecture Method		
TIM	Tivoli Identity Manager		
U2B	User-to-Business		
U2D	User-to-Data		
U2U	User-to-User		
UDB	Universal Database		
UDDI	Universal Description, Discovery, and Integration		
UML	Unified Modeling Language		
VCM	Value Chain Management		
VPM	Virtual Product Model		
VPN	Virtual Private Network		
W3C	World Wide Web Consortium		
WAP	Wireless Application Protocol		
WAR	Web archive		
WIP	Work in Progress		
WLM	Workload Management		
WML	Wireless Markup Language		
WS	Web services		
WS-I.org	Web Services Interoperability Organization		
XDE	eXtended Development Experience		
XHTML	Extensible Hypertext Markup Language		
XML	Extensible Markup Language		
XQuery	An XML-based method for querying databases		

Glossary

Access phase The first stage of the [e-business adoption cycle](#). In this phase, the Internet transformed business communication. E-mail replaced snail mail. Simple database queries—like checking a bank account balance or flight arrival information—superseded telephone calls and standing in lines. All that businesses needed to compete at this stage was a browser, Internet access and a simple Web site. This would change in stage two, the [integration phase](#).

Architecture Definition of the structure of IT solutions to business problems

Autonomic Like the body's autonomic nervous system, acting automatically or involuntarily, as by reflex. The same way we take for granted our body's management of breathing, digestion and the immune system, we will one day take for granted the ability of [autonomic computing](#) systems to manage, repair and protect themselves.

Autonomic computing One of the four characteristics of the [on demand operating environment](#). Computing systems will rapidly become too complex for humans to manage effectively—or to configure, manage, secure, optimize or repair. The solution? Technology that manages itself with a minimum of human involvement—similar to our own autonomic nervous system.

Business advantage A business advantage allows a company to move faster and travel farther than the competition in pursuit of its business goals.

Business analytics One of the five focus areas of [Grid computing](#), Business analytics enables faster and more comprehensive business planning through the sharing of data and computing resources.

Business intelligence Consolidates and analyzes raw business data and turns it into conclusive, actionable information. Enables companies to tap into disparate sources of customer, operational and market data—and then use this information to gain a competitive edge. It provides companies with the intelligence needed to spot trends, enhance relationships, reduce financial risk and create new sales opportunities.

Business integration Strategies and technologies to link people, processes and applications for maximum efficiency and effectiveness.

Business-to-Business (B2B, B-to-B) Business-to-business electronic commerce covers a broad range of business activities. For example, B2B systems exchange business documents, such as purchase orders and invoices, between partners in a supply chain. They implement a virtual marketplace (e-marketplaces or exchanges) wherein a single large manufacturer can consolidate the purchase of the goods that it sells in its stores. Marketplaces can also become trading marts or exchanges—either for commodity products, or for the range of products of a given type or industry segment. B2B systems also automate the purchase of goods that support a business's maintenance, repair and operation (MRO).

Business transformation The evolution of a company's processes and systems to become more efficient, effective and competitive.

Collaborate, collaboration Collaboration software—like that offered by Lotus—increases human productivity, bringing people together with messaging, calendaring, scheduling and other applications.

Content management Content management solutions control and integrate all forms of information—from images, documents and e-mail to Web pages, audio, video and more. As organizations strive for operational efficiency and cost-effectiveness in an increasingly on demand environment, content management technology becomes critical in solving business problems and gaining the competitive advantage.

Customer relationship management (CRM) Customer relationship management (CRM) strategies allow for the structuring of an entire enterprise around consumer wants and needs. CRM systems typically use technology and processes to get a more complete understanding of the customer, and to more rapidly deliver products and services around that understanding.

Data mining Data mining applies advanced algorithms to uncover trends and patterns in data that remain hidden to traditional analysis. Information for data mining is usually drawn from a [data warehouse](#).

Data warehouse A massive collection of raw data. [Data mining](#) is typically done on this collection to extract meaningful information.

Deep computing Ultrafast computing that is combined with sophisticated software to enable organizations to analyze, find patterns in and take action on the data that they have gathered.

Dynamic workplaces Online collaborative environments that provide flexibility and security for long-distance learning, meetings and more. Lotus Web conferencing offers a broad set of intranet-based collaborative technologies such as eHR, e-meetings, e-learning and instant messaging that can help companies reduce costs and make employees more nimble.

e-business The process of using Web technology to help businesses streamline processes, improve productivity and increase efficiencies. Enables companies to easily communicate with partners, vendors and customers, connect back-end data systems and transact commerce in a secure manner.

e-business adoption cycle The three-stage evolution of [e-business](#), which has led to today's on demand world. The three stages of this cycle are the [access phase](#), the [integration phase](#) and the [e-business on demand phase](#).

e-commerce The ability to buy and sell products and services over the Internet. Includes online display of goods and services, ordering, billing, customer service and all handling of payments and transactions.

e-marketplace Business communities are beginning to form e-marketplaces, enabling them to automate and leverage transactions with one another as a community. By bringing together large numbers of buyers and sellers, e-marketplaces give sellers access to new customers, expand the choices available to buyers and reduce transaction costs.

Engineering and design One of the five focus areas of [Grid computing](#), Engineering and design solutions allow companies to share data and computing power for calculation-intensive engineering and scientific applications, as well as to accelerate product design.

Enterprise optimization One of the five focus areas of [Grid computing](#), Enterprise optimization allows companies to get the most from their computing and data assets in order to improve utilization, efficiency and business opportunity.

Enterprise resource planning (ERP) Integrates processes and information across a company to accelerate customer service and deliver products faster. ERP applications, which “run the business” and are of primary importance to most companies, include essential applications such as financial accounting, asset management, billing, cost accounting, logistics, production planning and inventory control.

e-procurement The acquisition of direct and indirect products and services using the Internet and new technologies to facilitate a seamless, end-to-end stream of strategic procurement activities by connecting buyers with suppliers. Typically includes tools and business intelligence systems that enable improved [responsiveness](#) and analysis within the procurement organization.

e-sourcing e-sourcing can entail everything from server hosting to outsourcing an entire [IT infrastructure](#). It's a smart way for companies to stay focused on their core businesses while leveraging the expertise, people and cutting-edge technology of an infrastructure specialist like IBM.

Flexible, flexibility The flexible organization can transform itself quickly and easily when needed, to better adapt to changing business conditions and requirements.

Focused One of the four characteristics of an [on demand business](#). This business will concentrate on its core competencies, differentiating tasks and assets, while tightly integrated strategic partners manage selected tasks outside those core competencies—everything from manufacturing, logistics and fulfillment, to HR and financial operations.

Globalization As companies increasingly look to compete online and in the international marketplace, globalization—transforming business processes and strategies to better connect with customers, partners and suppliers around the world—is rapidly becoming a business requirement.

Globus Project™, The The Globus Project is a multi-institutional [research and development](#) effort, centered at Argonne National Laboratory, creating fundamental technologies for [Grid computing](#).

Government development One of the five focus areas of [grid computing](#), government development creates large-scale IT [infrastructures](#) to drive economic development and/or enable new collaborative government services.

Grid computing A type of distributed computing in which a wide-ranging network connects multiple computers, whose resources can then be shared by all end-users; includes what is often called "peer-to-peer" computing.

Grid computing allows companies to deliver computing power where they need it, only when they need it; they pay for what they use, when they use it. Companies have the choice to build their own grid (insource it) or tap into one built by IBM (outsource it).

Grid computing encompasses five focus areas: [research & development](#), [engineering & design](#), [business analytics](#), [Enterprise Optimization](#), and [Government Development](#).

Heterogeneous systems Networks that include computers and other resources from different manufacturers, using different standards. In the on demand environment, where all parts of a company are interconnected and existing investments in [infrastructure](#) cannot simply be discarded, heterogeneous systems are a necessity.

IBM intraGrid The IBM intraGrid, developed by IBM Research using [Globus Project™](#) technologies, brings together IBM R&D projects from around the world, and serves as a test bed for Grid services and solutions. Through the development and management of the intraGrid, IBM has gained valuable, first-hand experience in building and operating true commercial computing Grids.

Infrastructure Infrastructure includes all of the hardware, software and services that allow an [e-business](#) to function.

Integration One of the hallmarks of an [on demand business](#), and one of the four characteristics of the [on demand operating environment](#).

In the on demand world, integration goes far beyond connecting disparate computing assets, such as client and servers, so they can share information and "talk". The [on demand operating environment](#) must enable the integration of core business processes and systems so that business itself can flow inside and across enterprises. The very nature of these kinds of transactions demands data integrity, end to end—whether that end is in a government agency, a supplier, a distributor or a PDA in the hands of an individual consumer. Integrated systems are a key part of [business integration](#).

Integration phase The second stage of the [e-business adoption cycle](#). In this phase, [e-business](#) moved from access to transactions. The "information highway" was replaced by "e-business". Banks enabled customers to move money among accounts. Airlines let them make online reservations. And consumers could trade stock, apply for loans, purchase goods and services, renew their driver's licenses, even get an education. This required far more than simple Web sites. It required behind-the-scenes integration of internal systems and business processes. This would pave the way for the next stage of the cycle, the [on demand phase](#).

Java A popular cross-platform programming language used to add functionality to a Web site. Can be used on any PC or Macintosh Java™-enabled computer.

Knowledge management (KM) The creation, storage and collaborative sharing of employee information within the business environment, to enhance an organization's efficiency, productivity and profitability. Supported by advanced information technology tools and methods.

Legacy systems A company's existing computing hardware, software and services. Typically, these represent a significant investment of the company's time and resources. In the on demand environment, that investment is protected, as legacy systems remain available and networked with the rest of the [e-business infrastructure](#).

Linux A version of [UNIX](#) developed in 1991 by Linus Torvalds that may be shared freely and runs on many hardware platforms. [Open-standard](#) technologies like Linux are one of the cornerstones of on demand computing.

Middleware Software that connects two separate applications and passes data between them. For example, there are several middleware products that link a database system to a Web server. These allow users to request information from the database using forms displayed on a browser, and enable the server to return dynamic Web pages based on the users' requests and profiles.

Nanotechnology A field of science that aims to develop new atomic- and molecular-scale structures and devices for enhancing information technologies, as well as to discover and understand their scientific foundations.

Leaders in the development of nanotechnology, IBM's scientists have made numerous breakthroughs in the study of these small-scale devices.

On demand business A company whose business processes—integrated end-to-end across the company and with key partners, suppliers and customers—can respond with flexibility and speed to any customer demand, market opportunity or external threat. An on demand business has four key attributes: it is [responsive](#), [variable](#), [focused](#), and [resilient](#).

On demand operating environment The new computing architecture designed to help companies realize the benefits of [on demand business](#). The on demand operating environment has four essential characteristics: it is [integrated](#), [open](#), [virtualized](#), and [autonomic](#).

On demand phase The third phase of the [e-business adoption cycle](#), on demand represents a fundamental change across processes and enterprises. This phase will bring a totally new kind of transformation—or, more specifically, new levels of integration: of processes and applications inside the business; of suppliers and distributors at either end of the business; of customers outside the enterprise; of employees inside it.

Open Grid Service Architecture (OGSA) Open Grid Service Architecture is a distributed interaction and computing architecture based around the [Grid Computing](#) service, assuring interoperability on [heterogeneous systems](#) so that different types of resources can communicate and share information.

Open, openness, open standards One of the four characteristics of the [on demand operating environment](#). In this highly [integrated](#) environment, there are really only two choices: either everybody uses the same technology, or all technologies can connect and integrate. Open technical interfaces and agreed-upon standards are the only realistic way all of his can connect.

Open system A system whose characteristics comply with standards made available throughout the industry and that therefore can be connected to other systems complying with the same standards. This openness allows computing resources to work together, linking with devices and applications across organizational and geographic boundaries—an essential element of [on demand business](#).

Operating environment An operating environment consists of the system software and key applications that allow a computer or network to function.

Outsourcing A company chooses a consultant or application service provider to manage components of its internal IT structure, staff, processes and applications. This allows the organization to remain [focused](#) on its business goals.

Pattern Describes a proven solution to a recurring design problem, placing particular emphasis on the context and forces surrounding the problem, and the consequences and impact of the solution.

Pervasive computing The increasingly popular use of intelligent, easily accessible, sometimes invisible computing devices that simplify personal and business transactions. Designed to bring [open standards-based](#) applications to everyday life, these devices are most often mobile in nature, or embedded in the environment. Pervasive computing provides easy access to data stored on powerful networks, increases work efficiencies and removes much of the complexity inherent in these new technologies.

Product Lifecycle Management (PLM) Product Lifecycle Management allows companies to share common business processes and a common knowledge of a product by integrating all stages of its lifecycle, from concept to retirement, across the entire industry. Businesses can thus improve their [infrastructure](#), collaborate across the value chain and significantly reduce product development time and cost.

Portal A Web site or online service that provides a broad range of services (e.g., Internet searches or e-commerce) to a large audience.

Quality of service A term used in a Service Level Agreement (SLA) denoting a guaranteed level of performance (e.g., response times of less than one second).

Research and development (R&D) One of the five focus areas of [Grid computing](#). Companies can use grids to accelerate and enhance the R&D process by enabling the sharing of data and computing power seamlessly for research-intensive applications.

Resilient One of the four characteristics of an [on demand business](#). This business will manage changes and threats—from computer viruses, to earthquakes, to spikes in usage—with consistent availability, security and privacy.

Responsive One of the four characteristics of an [on demand business](#). This business will seem almost intuitive in its ability to sense and respond to dynamic, unpredictable changes in demand, supply, pricing, labor, competitors' moves, capital markets and the needs of all its constituencies—customers, partners, suppliers and employees.

Return on Investment (ROI) The income that an investment provides in a fixed amount of time. Also, the speed at which a capital outlay pays for itself. When companies make investments in Web technology, this measure is referred to as Return on Web Investment (ROWI). ROWI helps build a sound business case for e-commerce initiatives, applying a framework for analyzing and measuring the ROI of an e-business project at any point in the implementation.

Scalable, scalability Pertaining to the capability of a system to adapt readily to a greater or lesser intensity of use, volume or demand. For example, a scalable system can efficiently adapt to work with larger or smaller networks, performing tasks of varying complexity.

Self-configuring One of the characteristics of [autonomic computing](#). Self-configuring systems can adapt to dynamically changing environments with minimal human involvement.

Self-healing One of the characteristics of [autonomic computing](#). Self-healing systems can discover, diagnose and act to prevent disruptions, all with minimal human involvement.

Self-managing One of the characteristics of [autonomic computing](#). Self-managing systems can perform routine maintenance tasks with minimal human involvement.

Self-optimizing One of the characteristics of [autonomic computing](#). Self-optimizing systems can fine-tune resources and balance workloads to maximize use of IT resources, all with minimal human involvement.

Self-protecting One of the characteristics of [autonomic computing](#). Self-protecting systems can anticipate, detect, identify and protect against attacks, all with minimal human involvement.

Sense and respond One of the characteristics of [on demand business](#) is the ability to sense changes in the environment—fluctuations in customer demand, issues within the supply chain, variations in pricing structures—and respond quickly to meet those challenges.

Stages of adoption The three stages of adoption—[access](#), [integration](#), and e-business on demand—together make up the [e-business adoption cycle](#).

Standards-based The adoption of [open standards](#) allows for the [integration](#) of processes, applications and devices across the enterprise. Thus, standards-based computing is one of the cornerstones of the on demand environment.

Storage systems Hard disks and other systems designed for information storage and retrieval. In the on demand environment, these need to be both scalable and highly reliable.

Supercomputer Highly powerful, extraordinarily fast computers, typically used for research and other applications that require massive mathematical calculations. [Grid computing](#) gives e-businesses supercomputer-level power using their existing systems.

Supply Chain Management (SCM) An electronic alternative to the traditional paper chain, providing companies with a smarter, faster, more effective way to get the right product to the right customer at the right time and price. Combines the power of the Internet with the latest technology, enabling participating suppliers to access up-to-date company information and enabling companies to better manage and track supply and demand.

UNIX An [open-standard](#) operating system developed by Bell Labs in 1969. Systems running under UNIX (and UNIX variants such as [Linux](#)) can more easily be integrated with other resources in the enterprise to create an on demand computing environment.

Usage-based pricing One of the characteristics of [utility computing](#), usage-based pricing allows [on demand businesses](#) to pay for computing power as they need it, and only for as long as they need it, thus increasing business flexibility and resilience.

Utility computing One of the features of the on demand environment, utility computing delivers computing power to companies as they need it, and only for as long as they need it—making it as reliable, affordable and easy to use as common utilities.

Variable One of the four characteristics of an [on demand business](#). This business will use variable cost structures and adapt processes flexibly. This flexibility will enable it to reduce risk and to do business at high levels of productivity, cost control, capital efficiency and financial predictability.

Virtualized One of the four characteristics of the [on demand operating environment](#). The entire data center of an [on demand business](#) can be virtualized with an emerging technology called [grid computing](#). Like the Internet, Grid is based on [open technical standards and protocols](#). When implemented, grid allows a collection of distributed computing resources to be shared and managed as if they were one large, virtual computer.

Web Intermediaries A framework and set of programming tools from IBM for the uniform creation and control of intermediary programs such as proxy servers, transcoding processors, and any kind of program that sits somewhere between two end points in a network. Some other kinds of intermediary programs that can be built using WebSphere Business Integration include personalization of Web content; transcoding HTML for formatting to a handheld device; interactivity with other Web users and data; the filtering of content; and, more controversially, the monitoring of individual usage.

A WebSphere Business Integration application consists of a request editor, a (response) generator, a response editor, and a monitor. A collection of such a monitor, editors, and generator is known as a MEG, and a MEG constitutes an installable plugin. Plugins are registered in a computer and made usable whenever they are needed or wanted.

The (Java-based) WebSphere Business Integration Development Kit comes with some ready-made plugins, including the same plugin APIs as IBM's WebSphere Transcoding Publisher.

Web services A way of providing computational capabilities using standard Internet protocols and architectural elements. For example, a database Web service would use Web browser interactions to retrieve and update data located remotely.

Wireless Pertaining to communication that typically occurs over radio frequencies. See [pervasive computing](#).

Workload balancing Techniques that distribute the computing power needed to complete a task across multiple systems, so that no one computer is overwhelmed.

XML eXtensible Markup Language. A streamlined version of SGML, XML is regulated by WC3 (the World Wide Web Consortium). XML can create more advanced links than HTML.

Index

Symbols

.NET

Framework 290
platform 306

Numerics

3d com viii, 220, 229, 232, 248
portlet 238

A

A Portal Composite Pattern Using WebSphere V4.1 118, 129, 139, 149, 155
ABLE
See Agent Based Learning Environment
abstracts document 14
Access
Customer
Preferences use case 115
Service Content use case 173, 175, 178, 181, 195
eHR
System use case 195
use case 196
Engineering Data to Build Design Context
delta use case 215
Enterprise Applications use case 91, 174
Forum Information use case 115
from PDA use case 195
Help Desk use case 195
HR functional subsystem 178, 181
Integration
(Inspect Cargo) 138
delta 135
pattern 35, 75, 111, 113, 126
Pervasive Device Access 135
to Enterprise Applications use case 214
to Intranet and Syndicated Content
delta use case 215
to Multiple Business Processes
delta use case 215
access
control 45, 406

management 194
phase 61
point information and services 15
to digital information 59
accessibility 40
account
management 278
revocation 376, 382
ACL 164
Action 412
ActionForms 412
ActionForward 412
ActionListener 307
ActionMapping 412
Active Server Pages 305
ActiveX
controls 306
activities 65–66
actor
definition 86
ad hoc
access 19
management 209
ADA
See American Disability Act
Adaptive Differencing technology 379
adding integration server support 187
administration and security 24
Aerospace industry 201–203, 222, 248
key business drivers 208
affinities 403
processing 313
Agent Based Learning Environment 350
Aggregator 139
Alert Management 143
alphaWorks 290, 293
autonomic zone 351
American Disability Act 396
analytic applications 14
ANSI SQL/MED standard 325
AOD 70, 97, 99–101
(also see Architecture Overview Diagram)
(Logical Nodes) 97, 128–131, 133–134
delta patterns

Access Integration
 (Inspect Cargo) 135

Application Integration
 (Assess Risk) 135
 (Generate Alert) 135
 (Generate Incident) 135

Extended Enterprise
 (Find Agency Information) 135

use cases
 Request Entry 140
 User Management 140

(Subsystem) 97, 99–101

Applied Pattern
 Access Integration
 Pervasive Device Access 126

Application Integration
 Broker 126
 Direct Connection 126

Extended Enterprise
 Exposed Business Service 126

delta requirements
 Access Integration
 (Inspect Cargo) 126

Application Integration
 (Assess Risk) 126
 (Generate Alert) 126
 (Generate Incident) 126

Extended Enterprise
 (Find Agency Information) 126

(subsystem) 179

associated use cases
 Assess Risk 99, 108, 130
 Authenticate and Authorize User 99
 Author and Approve Bulletin 99
 Establish Team and Environment 99
 Find SMEs 99
 Generate Alert 99, 108
 Inspect Cargo 99
 Manage and Track Incident 99
 Publish Bulletins 99
 Register with Agency 99
 Request Entry 99, 105, 108
 Resolve Incident 99
 User Provisioning 99

functional
block
 Entry Management System 104–105,
 117
 Inspection System 117

subsystems
 Cross-Agency
 Collaboration 99
 Document Management 99
 Information 99
 Entry Management 99, 108–110
 Inspection 99
 Risk Management 99, 108–109
 User Management 99, 130

laying out key points 99

Logical Node view 128

seeding the Operational Model with 150

Subsystem view 128

use cases
 Find Agency Information 99
 work product 69, 81, 97–98, 103–105,
 108–114, 116–118

Apache
 Axis 301
 project 290
 HTTP server 422
 Jakarta project 310
 SOAP 290, 301
 Tomcat 290
 Web server 288
 XML project 290

applets 272
 Form Creation 277
 Workflow Design 277

Application
 Integration
 (Assess Risk) 137
 delta 135
 (Generate Alert) 139
 delta 135
 (Generate Incident) 139
 delta 135
 pattern 35, 109, 111, 117, 126, 137, 139,
 176
 Broker 127, 135, 137
 Business 108
 Direct Connection 127, 135, 137
 Managed Process 127, 137
 Transactional 127, 137
 patterns 30, 38, 69–72, 76–77, 97, 109–110,
 112, 118–121, 123, 142, 154
 definition 32
 Document Exchange 125
 Exposed

Application 125
Business Services 125, 130
for Access Integration 94
Managed Public
and Private Process 125
Process 125
mappings 97
Runtime pattern 70
selecting 38
Server 139
logical node 143
Web 282–283

application
and workflow 24
back-end integration 18
complexity 18
defining the network structure 38
design and development guidelines 31
integration 107
portlet builders 417
sharing 403

Apply for Job
business/integration pattern 184
use case 173–174, 178, 181

Approve ECR delta use case 214

Arabic 309

architect 30

architectural decisions 95, 97

Architecture
Decisions
document 97, 127, 151
Alternatives 116, 127, 136–139
Architectural Decision 116, 127, 135, 137–139
Assumptions 116, 127, 135, 137–139
Decision 116, 127, 136–139
Derived Requirements 116, 127, 136–139
Implications 116, 127, 136–139
Issue or Problem 116, 127, 135, 137–139
Justification 116, 127, 136–139
Motivation 116, 127, 136–139
Related Decisions 116, 127, 136–139
work product 69–70, 81

Overview Diagram 69
(Logical Nodes) 151
(subsystem) 177

Also see AOD

Architecture Overview Diagram
(also see AOD)
Ariba 288, 291, 406
artificial intelligence 287
ASP 305, 307
.NET 290
pages 306
See also Active Server Pages
Assess Risk 99, 108, 130
application integration pattern 118, 126–127, 137
extended enterprise pattern 118, 135
delta 135
function 133
use case 88, 102, 121, 133
assessment offerings 9

assets
customizable 4
reusable 4, 66–67
candidate 68

AST
See automatic summary table

asynchronous messaging 275

Attach Documents to Workflows delta use case 215

Authenticate
and Authorize User 99
use case 87, 99, 102, 113, 173–174, 177, 181
use case 91
authentication 45, 95, 218, 228, 278, 391, 406
credentials 164

Authentication use case 214

Author and Approve Bulletins
use case 87, 92, 102, 113

authorization 96, 218, 228, 278, 391

Authorize
and Authenticate User use case 91
use case 91, 214

automatic
summary table 333

automation 11, 272, 346

Automotive industry 201–203, 209, 222, 248
key business drivers 208

autonomic
activity 369
capabilities 7, 373
computing 8, 257, 261, 265, 271, 346, 348–349, 351, 355, 359
architecture concepts 365

- customer value 362
 grid infrastructure 355
 in the IT environment 367
 levels 369
 systems 269
 computing-managed element 360
 cycle 351, 360
 elements 351
 IT resources on a grid 271
 manager 366
 on demand operating environment 59
 operational environment 208
 solutions 350
 technologies 270
Autonomic Computing Toolkit 352
 autonomics 362
Autonomy 389
 availability 11, 45, 57, 81, 95, 218
- B**
- B2B** 13–14, 26
 marketplaces 45
 pattern
 See **Business-to-Business pattern**
 portals 41, 396
 Web applications 391
B2C 13, 26, 414
 e-business systems 22
 portals 41
 Web applications 391
B2E 13, 26
 e-business 22–23
 systems 22
 enterprise applications 414
 implementation 26
 portals 62, 167, 174
 employee
 See **employee portals**
 knowledge worker
 See **knowledge worker portals**
 types 41
 backup and recovery 57, 81
 tools
 centralized 381
 backuparchive client 379
 bandwidth
 reduced cost 270
 base model 157
 baseline solution 93
Basic
 authentication/authorization 96, 163
 Profile 295
BCS 56
BEA
 application server 306
 portal server 306
bean 306
benchmarking expectations 247
best practices 4–5, 57, 65
 definition 55
BI 44
 portals
 definition 14
bi-directional
 languages 309, 396
 translation of content 396
Bill
 of Lading
 service 143
 of Materials 205
Binding Template 293
BOL
 See **Bill of Lading**
BOM
 See **Bill of Materials**
bottleneck 160
BPA
 See **Business Process Automation**
BPEL4WS 291
 (also see **Business Process Execution Language**
 for Web services)
BPM 212, 225
 (also see **Business Process Management**)
branding 44
Brazilian Portuguese 397
brick-and-mortar businesses 20
Broker application integration pattern 126–127, 135, 137
brokered cooperation 387
Browse and Select from Catalog use case 115
buddy lists 403
Business
 Continuity and Recovery 12
 Entity 293
 Intelligence 274
 patterns 29, 31–32, 34–36, 69, 71–72, 76, 89,

- 92–94, 97, 103, 105–107, 109–110, 117, 142, 154, 414
 - Application Integration
 - See Application Integration pattern
 - patterns 111
 - Directly Integrated Single Channel 110
 Collaboration
 - See Collaboration business pattern
 definitions 32–33
 delta 69, 117–118, 125
 Extended Enterprise
 - See Extended Enterprise business pattern
 Information Aggregation
 - See Information Aggregation business pattern
 mappings 97
 selecting 38
 Self-Service
 - See Self-Service business pattern
 Process
 - Automation 202, 215
 - Execution Language for Web services 291
 - Management 212, 218–219, 225, 242
 Service 293
 business 104
 domain 30
 drivers 30–31, 51–54, 76, 81, 103–104, 110, 118–121, 130
 - core 16
 - portal 16
 events
 - reduced latency of 53
 - reducing latency 220
 functions 80
 intelligence 14
 logic 48, 95
 operations 96
 partners 12, 200
 patterns 103
 portal 15
 process integration 109
 processes 5, 7
 protocols 125
 rules 44, 89
 strategy and roadmap 81
 business-oriented requirements 94
 business-to-business
 - See B2B
 Business-to-Business pattern
 - See Extended Enterprise business pattern
 business-to-consumer
 - See B2C
 business-to-employee
 - See B2E
 byte streams 342

C

- C++ 425
 caching 160
 CAD 205, 210, 238, 249
 - (also see Computer Aided Design)
 calendaring 14, 423
 calendars 43–44
 Calypso 339
 CAM 205, 210
 - (also see Computer Aided Manufacturing)
 capabilities
 - autonomic 7
 - self-healing 7
 capacity
 - planning 57, 81
 Cascading Style Sheets 308–309
 case study 78, 117
 AOD 152
 - (Subsystem) 112
 Application patterns
 - Directly Integrated Single Channel 113
 - Personalized Delivery 113
 Population
 - Crawl and Discovery 113
 - Single Step 113
 Store and Retrieve 113
 Web Single Sign-on 113
 business drivers 83, 121
 IFB 168
 - (also see IFB case study)
 IT drivers 83, 89
 list of actors and existing systems 85
 major functionality needed 85
 NFRs
 - availability 95
 - data integrity 96
 - maintainability 95

manageability 96
performance 95
security 95
system usability 96

Portal composite pattern
process used to conclude applicability 103

Project Description 81

System Context Diagram 83

use cases 86, 99
 Authenticate and Authorize User 113
 Author and Approve Bulletins 113
 Find
 Agency Info 113
 SMEs 113
 Inspect Cargo 113
 Manage and Track Incident 113
 Publish Bulletin 113
 Register with Agency 113
 Request Entry 113
 Resolve Incident 113
 User Provisioning 113
 vision 83

catalog
 See Knowledge Map

CATIA viii, 235

cell
 managers 356
 phones 45

central user directory 139–140

centralized administration
 user identities, credentials, and permissions 391

CGI
 (also see Common Gateway Interface)
 scripts 305

change
 control 96
 impact analysis 365
 management 206, 364
 process 222, 242

channels
 multiple delivery integration across 53

chat 14, 43–44, 214, 228, 251, 253–254, 404, 422, 424
 rooms 41
 simple via IRC 113

Chinese 309, 396

Choose Financing Option use case 115

cHTML 406

CICS 343

CIS
 See Customer Information Service

click-to-action
 portlets 387
 technology 418

client
 programming model 303
 proxies 293

clipping portlet 25

Cloudscape Java database 418

cluster services 24

CODASYL 335

Codd's relational model 335

Collaboration
 business pattern 33, 75, 113
 Center
 See Lotus Collaboration Center

collaboration 23, 41, 149
 guidelines 155
 portals 43
 portlets 401
 server 188
 services 47, 384

collaborative
 environment 50
 portals 13, 15
 definition 14
 portlets 424
 services 12–13, 43–44
 tools 211, 213
 toolsets 222
 workplace 12

commerce
 case study use cases
 Access
 Customer Preferences 115
 Forum Information 115
 Browse and Select from Catalog 115
 Choose Financing Option 115
 Notify Customer 115
 Perform Account Maintenance Functions
 (user provisioning) 115
 Place Order 115
 Query Inventory 115
 Register with Company 115
 Send Order to Warehouse 115
 portals 18
 commerce-enabled portals 18–19, 406

Common Gateway Interface 305
(also see CGI)
Common Object Request Broker Architecture 392
communication
 portlet-to-portlet 47
 services 44, 47
communities of interest 47
complete portal solution 13
Component
 Interaction Diagram 144–148, 150, 191
 Model 72–73, 128, 142, 151
 Relationship Diagram 144
 work product 71, 98, 144
 Relationship Diagram 190
component 143
 design 95
 interactions 71
component-based development 28
Composite patterns 29–30, 34–36, 68, 72, 75–76, 89, 97, 101, 107, 110–113, 142, 154
 definition 32
e-Marketplace
 See e-Marketplace composite patterns
mappings 97
Runtime pattern 69
Sell-Side Hub
 See Sell-Side Hub composite pattern
types 37
using to identify Application patterns 110
Computer Aided
 Design 210
 Manufacturing 210
Conceptual Operational Model 152
connections 57
 management 159
 pooling 159
connectors 24
Consolidated View delta use case 215
constraints
 configuration 95
 operational 95
 physical 95
 standards 95
 system 95
 technology 95
consultant partners 12
consumer portal 15, 45
content
 aggregation 46, 50
caching 47
delivery services 160
management 13, 16, 43, 50, 89, 414
 guidelines 149–150
repositories 390
services 44, 345
support 47
 system 143, 334, 337, 339, 341
managing 218, 228
personalized view 218
providers 388
publishing 228
cooperative portlets 418
CORBA
 (also see Common Object Request Broker Architecture)
 credentials 392, 394
core competencies 6, 58
Corporate Collaboration
 functional subsystem 178, 181
 use case 174–175, 178, 181
COTS portlet 25
Create
 and Deploy Business Processes
 delta use case 215
 and Manage Business Object
 delta use case 215
 Content use case 91
 ECR delta use case 214
credential-based authentication 305
credit verification and authorization 247
Critical Success Factors 165
CRM 14, 22, 41, 274, 404
 functionality 96
 portal 15
Cross-Agency
Collaboration
 functional subsystem 99, 102
 Server 135
 logical node 143
Document Management
 functional subsystem 99, 102
 logical node 143
Information
 functional subsystem 99, 102
cross-selling 17, 19
 effective 53
CrossWorlds 249
CSV file 281

cultural differences 76
Current System Context work product 85
custom
 design 31, 35–36
Customer
 Information Service 159
 Relationship Management
 See CRM
 Service functional subsystem 178, 181
customer
 care system 369
 fulfillment 7
 intimacy 51
 knowledge 50
 mindshare 50
 retention 51
 satisfaction validation 247
Customizable Reports and Audit Trails
 delta use case 215
customization 44, 53, 57
Customize Home Page use case 91
Customized Presentation to Host
 self-service application pattern 221–222
Customs Inspection System 143

D
Dassault Systèmes viii
data
 aggregation 50
 components placement 81
 integration 107
 back-end enterprise 96
 marts 332
 mining 14, 43–44, 273, 339
 warehouse 210, 337, 340
 distributed 332
 virtual 332
DataMonitor 388
DB2 228, 269, 355, 390–391, 405, 425
 DataJoiner 322–323
 DataPropagator 331
 Enterprise Edition 422
 Extenders 337
 Information Integrator 257, 320, 346, 390
 crawler and text search features 390
 federated search functions 391
 for Content 390, 425
 Relational Connect 322
Spatial Extender 332
SQL 333
 capability 324
UDB 274, 322
 (also see DB2 Universal Database)
Universal Database 257, 274, 320, 346
XML Extender 275

DBMS
 evolution of architecture 336
 evolution of technology 335
decision support system 273
deep
 analysis 338, 340
 customer knowledge
 and mindshare 16, 51, 89
delta
 patterns
 Access Integration (Inspect Cargo) 118
 Application 126, 130
 Integration
 (Assess Risk) 118, 127
 (Generate Alert) 118
 (Generate Incident) 118
 Runtime pattern 70
 Business 69, 117–118, 125
 Extended Enterprise
 (Assess Risk) 118, 135
 (Find Agency Information) 118
 Integration 69, 117–118, 125
 requirements 68, 72, 117
 AOD 69
 product mappings 153
 use cases 68, 93
 classifying 94
demand intelligence 14
denial-of-service attacks 379
deployment flexibility 62
design
 logical 81
 system 97
 physical 81
destroy method 302
developers 31
developerWorks 347, 351, 355, 359
 Java zone 272
 WebSphere
 Portal Zone 308
 Studio Zone 307
developmentor 290

digital
asset management 344
content libraries 389
signature 305

DII 304
(also see Dynamic Invocation Interface)

Direct Connection 139
application integration pattern 126–127, 135, 137

Directed Collaboration collaboration pattern 196

Directly Integrated Single Channel
application pattern 113, 184

Directory
and Security Services node 140
Services Markup Language 279

Discovery Server 312–313, 315–318, 321
API Toolkit 319
portlet 402

discussion groups 14, 41, 44
material 13

Display Home Page use case 91

distributed
computing systems 10
data warehouse 332
programming 290

DMU 242
visualizing 229

DMZ 140, 158, 196

Do Online Training use case 195

Document
Exchange application pattern 121, 125
Manager 416
Web 281

document-sharing via TeamRooms 113

Documentum 322, 388, 391, 406

doEdit requests 307

doGet requests 307

domain
functional 80
operational 81

Domino
See Lotus Domino

Domino.doc 390
libraries 425

doPost requests 307

double-byte languages 309, 396

doView requests 307

Dreamweaver 389

DSML 279–280

(also see Directory Services
Markup Language)
connectivity 280

DTDs 344, 422

Dynamic
Invocation Interface 304
Workplace 213
advantages 229
composite pattern 223
Portal 189
composite pattern variant 223

dynamic
content 45, 48
discovery of available services 387
programming 327
proxy 304
Web content 408
workflow 253
workplace 26

E

EAI 24, 228, 249
(also see Enterprise Application Integration)
process-centric 212, 225
standard framework 212
work-centric 212, 225

EAR 298, 300
(also see Enterprise Application Archive)

ease of use 53

e-business 3, 426
advanced stage 60
applications 12
models 60
on demand 5, 10–11, 58, 259
evolution 59
phases 262
recommended products 273
return on investment 263
stages 261

portal 19

solution entities
data that exists within the organization 103
interacting enterprise/organization 103
users of the solution 103

strategy 77

EBXML 339

Eclipse 273, 310, 320, 346

e-commerce 107, 274, 422

portal 15
standalone Websites 15
ECOs 207, 215, 230, 240–242, 244–245, 248
(also see Engineering Change Orders)
ECRs 206–207, 214–215, 225, 230, 236, 239–240,
244–245, 248, 250
(also see Engineering Change Requests)
process 216
request 220
EDI 125
EDS viii
e-government 117, 136
case study
(also see case study)
AOD 153
product mapping 156
associated system components
Alert Management 143
Content Management System 143
Customs Inspection System 143
Emergency Response Team
Environment 143
Entry Management 143
Health Department Portal 143
Integration Hub 143
Lookout ODS Service 143
Port Authority Portal 143
Remote Document Search Service 143
Risk Assessment System 143
Component
Interaction diagram 145–148
Model Relationship Diagram 144
logical node components
Application Server 143
Cross-Agency
Collaboration Server 143
Document Management 143
Entry Management 143
Existing Agency Servers 143
Inspection System 143
Integration Hub 143
Message Broker 143
Presentation Server 143
Operational Model 157
with high-availability NFR walkthrough
162
characteristics 78
key initiatives 79
services 96

System Context Diagram 84
Use Case
descriptions 87
Model 87
use cases
Assess Risk 88, 121, 133
Authenticate and Authorize User 87
Author and Approve Bulletin 87
Establish Team and Environment 88
Find
Agency Information 88
SMEs 88, 115–116, 127
Generate
Alert 88
Incident 88
Inspect Cargo 87
Manage and Track Incident 88
Publish Bulletin 88
Register with Agency 88
Request Entry 88
Resolve Incident 87
User Provisioning 88
vision 80
work product
AOD (Subsystem) 105–106
EIP 322
(also see Enterprise Information Portal)
EJB 295–296, 414, 422
(also see Enterprise JavaBean)
engine 358
implementation 302
service implementation 302
stateless session deployment 302
ejbCreate method 302
ejbRemove method 302
e-learning 168
e-Learning System
business/integration pattern 184
electronic auction system 369
Electronic Commerce composite pattern 113–114
e-mail 13, 43, 113, 214, 236, 251, 253–254, 312,
319, 390, 404, 422–424
distribution lists 337
integration 228
notifications 215
servers 272, 290
e-Marketplace composite pattern 38
e-meetings 14, 232, 236, 423–424
integration 228

Emergency Response Team Environment 143
Emerging Technologies Toolkit 351
Employee
e-learning use case 174–175, 178, 181, 195
Self Service 169
employee
portals 15, 41
retention 169
employee-to-employee communication 169
Energy industry 248
Engineering
Change
Management processes 244
Order 207, 244
Request 207, 210, 214, 244
Requirements 222
Release Repository 232
English 396, 423
ENOVIA 221
3d com 221
LCA 249, 253
PDM systems 221
PPR viii
products 221
VPM viii, 220, 225, 232, 237–238, 249, 253
ENOVIA 3d com 229
enrollment system 45
Enterprise
Application
Archive 298
Integration 225, 243
Information Portal 322, 425
JavaBean 275, 296
Resource
Management 62
Planning 107, 232, 348
enterprise
application
integration 384
systems 334
integration phase 61
portals 45
security
manager 140
model 305
systems
integration with 15
Enterprise Java Beans 422
Enterprise-wide
Process Change Management 201
Resource Planning 211
entrance 39
Entry
Management 143
application 141
functional subsystem 99, 102, 105
logical node 143
System
functional block 104, 109–110, 117
Request Form 141
Environmental Systems Research Institute 332
Envisioned Goals and Issues
document 81, 154, 165
work product 80
ePCM ix, 201–202, 228, 237, 239–240, 248
(also see Enterprise-wide Process Change
Management)
architecture 252
automatic ID creation 244
Business Process Manager 232
Change Management processes 245
Collaborative Suite 251
Document Manager 244
Viewer Center 249
EAI framework 249
ECR 238
workflow 237
Engineering Manager 238
form templates 244
functionality 220
IT processes 244
key
benefits 253
features 244
Managed Process
application integration patterns 218
modules
defined 253
Object
Manager 248
Order Handling Process 247
portlet 237
process
model 244
modeling 244
templates 244
Process Monitoring and e-Reporting 249
Procurement and Materials 244

Quality Management processes 244
reject loop support 244
role-based authorizations 244
standard templates 244
Task Allocation processes 244
templates 229
 for engineering processes 245
 for order processing 247
 for the Work Management
 and Task Allocation process 246
VPM Connector 250
WIP
 notification 244
Work Management
 and Task Allocation template 246
 processes 244
workflow
 federation 248
 worklist templates 244
eQ 12
ERM
 See Enterprise Resource Management
ERP 107, 211, 214, 219, 228, 230, 248–249, 253, 274, 348, 404
 (also see Enterprise-wide Resource Planning)
ES
 See Lotus Extended Search
e-Server Integrated Platform for e-business 12
e-services models 96
e-sourcing 8, 270–271
ESRI
 See Environmental Systems Research Institute
Establish Team and Environment 99
 use case 88, 91, 102
ETTK 351, 355
 (also see Emerging Technologies Toolkit)
event notification services 47
e-workplace 5, 423
Excel 281
Excite 42–43
execute phase 57
Existing Agency Servers logical node 143
existing skills 18
Exposed
 Application
 application pattern 121, 125
 Business
 Process
 application pattern 138–139
Services
 application pattern 125, 130–131, 136
 Runtime pattern 132–133
 extended enterprise pattern 126, 135
Extended
Enterprise
 (Assess Risk) 135
 (Find Agency Information) 138
 delta 135
 patterns 126, 138, 176
 Application 133
 business 33, 112, 117, 131
 Application 119–120
 Exposed Business Service 135
 Single Sign-On access integration pattern 196
eXtended Development Experience 273
extended enterprise portal
 See extranet
extensibility 17, 54
Extensible Markup Language 272, 275, 282
extranet 41

F

Factiva 388
failover 163
failure identification 357
faster machines 159
FatWire 388
federated
 access 41
 database system
 when to use 330
 DBMS architecture 341
 portal 400
 query engine 343
 searches 391
 server 327
 system 323
 Web services and XML 346
federating
 existing enterprise applications 215
 workflows 225
federation technologies 322
Filenet 391
Financial
 industry 248
 Times 388
financial

applications 7
management 14, 22
systems 168

Find

- Agency Information
 - extended enterprise pattern 118, 126, 138
 - use case 88, 99, 102, 113
- SMEs 99
 - functional block 115, 127
 - use case 88, 92, 102, 113, 116

firewalls 163, 309

flexible wrapper architecture 343

focus 58, 208

Ford 202

form-based authentication 392

forms 13

fragment 46

frames 310

frameworks 4–5

French 396

FrontPage 389

functional

- block 98
 - (also see functional subsystems)
 - use cases 103
- domain 57
- requirements 70
- subsystems 71
 - (also see functional blocks)

Functional Requirements work product 84

future business goals 78

G

Galaxia ix, 249

- adding delta application patterns 223
- BPM
 - rules system 219
- change management process 219
- core competencies
 - Business Process Modeling 202
 - Integration 202
- delta Runtime patterns 223
- demo scenario 230
 - ECR process scenario 236
 - Phase 1 - Identify the change proposal
 - Approve ECR 237
 - Create an ECR 237
 - Exchange with client 236

Phase 2 - Approval Change Analysis

- Project Manager 240
 - Assistant 240

Phase 2 - Engineering Change Analysis

- Designer 238
- Engineering Manager 238

Phase 2 - Finance Change Analysis

- Purchaser 239
- Supplier 239

Phase 3 - Executing the change

- Committee 241
- Engineering, Manufacturing, Purchasing, Methods 241
- Project Manager 241
 - Assistant 241

Phase 4 - Final review and release

- Committee and Client 242
- Project Manager 242

interactions between various applications

- in the scenario 235

product mapping 232

ECR process 230

Engineering Change Requirements 214

ePCM

- See ePCM

PCM

- process 220
 - latency 220

PDM

- systems 220
- portal solution rationale 218

Production Release Design 205

solution 212

- Customized Presentation to Host
 - self-service application pattern 221–222
- delta use cases
 - Access
 - Engineering Data to Build Design Context 215
 - to Intranet and Syndicated Content 215
 - to Multiple Business Processes 215
 - Attach Documents to Workflows 215
 - Consolidated View 215
- Create
 - and Deploy Business Processes 215
 - and Manage Business Object

215
Customizable Reports
 and Audit Trails 215
Object Representation of Processes
 and Linking 215
Process/Activity Based Workflow
 Approve ECR 214
 Create ECR 214
 Manage and Schedule Work 214
Publish and Manage User Content 214
Synchronous Collaboration 214
Trigger/Control Application Level
 Workflows 215
Work Notifications and e-Mails 215
Workflow Monitoring 215
generic use cases
 Access to Enterprise Applications 214
 Authentication 214
 Authorization 214
 Manage Users 214
 Personalize Home Page 214
 User Registration 214
 View and Send e-Mail 214
Non-Functional Requirements 218
product mapping 226
 Runtime pattern 224–226, 228
Solution Overview Diagram 212, 216–218
U2B requirements 221
vehicle development
 process 204
 concept sorting and feasibility analysis
 activity 205
 development activity 205
 product description activity 204
 prototyping activity 205
 production release activity 205
VPM Connector 249
Garlic
 research project 323
 technology 322
gate 39
general public portals 42
Generate
 Alert
 application integration pattern 118, 126, 139
 use case 88, 102
 Incident
 application integration pattern 118, 126, 139
 use case 88, 102

Schedule
 business/integration pattern 184
 use case 173, 175, 178, 181
geospatial data 332
German 396
getting buy-in 56
GISes
 See geographic information systems
global
 communication 422
 process visibility 225
Global Unique IDs 419
globalization 203
Globus 269
Google 42
grid 6, 271
 computing 8, 10, 59, 257, 261, 265, 271, 355,
 357
 standards 358
 definition 269
 infrastructure for autonomic computing 355
 open protocols 269
 services 358
 structure 351
 technologies 270, 321
grid-like behavior 357
group scheduling 423
GUIDs
 See Global Unique IDs

H

HandleMap 358
hardware
 connections 81
 locations 81
 platforms 57, 81
 topology 81
Health
 Department Portal 143
 industry 248
heterogeneous replication 331
Hewlett-Packard viii
high-level functional requirements 77
HomePage Builder 389
Hoovers 388
horizontal
 portals 16, 63
 definition 15

framework 16
hosting environment 360
HP-UX viii
HR 22, 167
 functions 168
 systems 7
HTML 19, 46, 272, 281, 284, 306, 310, 389–390, 392, 406, 408, 422
 (also see Hypertext Markup Language)
 pages 275, 388
HTTP 281, 288, 357
 Basic Authentication 164
 BASIC-AUTH 305
 GET 289, 291, 307
 POST 289, 291, 307
 request 48, 164
 servers 422
HttpFormBasedCredential 393
HTTPS 280, 393
HttpServlet 386
HttpServletRequest object 305
HttpServletResponse object 305
HVWS 158
 (see IBM High Volume Web Site)
Hypertext
 Markup Language 272, 281
 Transfer Protocol 281

|

IBM 154, 288
 Autonomic Computing Initiative 362
 Content Manager 322, 344, 389–390, 426
 Directory
 Integrator 280
 Server 392, 422, 424
 Enterprise Information Portal
 See Enterprise Information Portal
 eServer products 268, 355
 federated database
 architecture 326–327
 technology 321
 Global Services 30
 High Volume Web Site 157, 161
 HTTP server 422
 IT Process Model 363
 Lotus Workplace Web Content Management 389
 Patterns for e-business

See Patterns for e-business
SOAP4J 290
software play 11
Systems Journal 295
systems play 11
Tivoli Monitoring family 368
TotalStorage solutions 355
WebSphere Edge Server 159
XML and Web Services
 Development Environment 293
IDE 310, 422
 (also see Integrated Development Environment)
identity management 277
IDI 281
 (also see IBM Directory Integrator)
IDL 290
 (also see Interface Definition Language)
IETF 353
IFB
 (also see International Food and Beverage)
AOD
 (logical nodes) 188
associated use cases
 Access Customer Service Content 178, 181
 Apply for Job 178, 181
 Authenticate and Authorize User 177, 181
 Corporate Collaboration 178, 181
 Employee e-learning 178, 181
 Generate Schedule 178, 181
 Job Solicitation 178, 181
 Manage
 Compensation and Benefits 178, 181
 Tasks 178, 181
 Receive and Audit Inventory 178, 181
Review
 and Guide Schedule 178, 181
 and Perform Tasks 178, 181
 Update Personnel Records 178, 181
 User Management 177, 181
functional subsystems
 Access HR 178, 181
 Corporate Collaboration 178, 181
 Customer Service 178, 181
 Inventory 178, 181
 Labor Scheduling 178, 181
 Recruitment 178, 181

Task Management 178, 181
Training 178, 181
User Management 177, 181
Identified software services 192
Architecture Overview Diagram
(subsystem) 179
case study
(also see case study)
business requirements 169
delta
Application patterns
Directly Integrated Single
Channel 184
Managed Collaboration 184
Pervasive Device Access 184
Router 184
Store and Retrieve 184
Business and Integration patterns
Apply for Job 184
e-Learning System 184
Generate Schedule 184
Job Solicitation 184
Manage Tasks 184
Review
and Guide Schedule 184
and Perform Tasks 184
User Management 184
requirements 182
IT requirements 169–170
project scope 170
System Context Diagram 171
Use Case descriptions
Access Customer Service Content 173
Apply for Job 173
Authenticate and Authorize User 173
Corporate Collaboration 174
Employee e-Learning 174
Generate Schedule 173
Job Solicitation 173
Manage
Compensation and Benefits 174
Tasks 173
Receive and Audit Inventory 173
Review
and Guide Schedule 173
and Perform Tasks 173
Update Personnel Records 174
User Management 174
company background 168

Component
Interaction Diagram 191
Relationship Diagram 190
core functions 168
logical nodes
associated software services 191
mapping
software 191
NFRs
Availability 176
Performance 176
Reliability 177
Response time 177
Security 177
Single Sign-On 194
Operational Model 192
walkthrough 194
Portal solution 168
Runtime pattern 196
solution development process 197
Use Case Model 173
use cases
Access Customer Service Content 175, 195
Apply for Job 174
Authenticate and Authorize User 174
Corporate Collaboration 175
Employee e-learning 175, 195
Generate Schedule 175
Job Solicitation 175
Manage
Compensation and Benefits 175
Tasks 175
Receive and Audit Inventory 175, 195
Review
and Guide Schedule 175
and Perform Tasks 175
Update Personnel Records 175, 195
User Management 175
iFrame portlet 25
IIOP 357
ilities 95, 218
IML processors 376
Implement phase 58, 165
IMSTM 335
incident management 364
indexing intelligence 14
industrial portals 42
industry solutions 68–69, 89
Industry Value Project 11, 56

inetOrgPerson schema 391
Information
 Aggregation business pattern 33, 75, 113, 115, 127
 Integrator
 advanced search functions 391
 Request (data integration) 139
information
 aggregation 40
 categorization 14
 integration platform 342
 islands 211
 management
 client 423
Information Integrator
 See DB2 Information Integrator
Informix 405
 DataBlades 337
 IDS 322
Infrastructure Solutions with Server Management 12
init method 302
Inktomi 389
iNotes portlet 417
Inspect Cargo 99
 access integration pattern 118, 126
 use case 87, 102, 113
Inspection
 Diagram 145
 functional subsystem 99, 102
 System
 functional block 117
 logical node 143
installable unit 360
InstallShield 350
instant messaging 232, 403, 423–424
Integrated
 Development Environment 310
 Industry Solutions 56
integrated
 content 12
 on demand operating environment 59
 operational environment 208
Integrated Development Environment 422
Integration
 Hub 143
 Assess Risk process 141
 patterns 29, 31, 35–36, 69, 71–72, 76, 89, 92–94, 97, 103, 108–109, 142, 154, 414
Access
 See Access Integration pattern
Application 111
 (also see Application Integration pattern)
applying 106
definition 32, 34
delta 69, 117–118, 125
identifying and applying 107
mappings 97
integration 6–7, 11, 24, 271
 back-end 62
 business 273
 data 274
 horizontal 273
 out-of-the-box 27
 point 108
 process 79
 server 187
 technical 79
integrator partners 12
Intelligent Miner
 for Text 344
 Scoring 344
interactive messaging facilities 80
Inter-Agency Business Services 135, 137–139
Interface Definition Language 290
intergovernmental linkages 80
International Food and Beverage 168
international language support 57
internationalization 309, 396
Internet
 Mail Box portlet 417
 news groups 417
 portals 13
 (also see public portals)
 protocols 288
 standards 288
inter-organization interaction 80
Interwoven 388
intranet
 portals 13, 41
 (also see B2B portals)
intrusion sensing 381
inventory
 control 115
 just-in-time 169
 processes 169
Inventory functional subsystem 178, 181
IPL 376

iPlanet 392
IRC 113
iSeries 27
issues discovery process 77
ISV
 partners 7, 12
IT
 Architect 98
 drivers 17, 30–31, 76, 81, 103–104, 110, 118–120, 122, 130
 Infrastructure Library 363
 infrastructure Workbench tools 373
Italian 396
ITIL
 See IT Infrastructure Library
IVP
 See Industry Value Project

J

J2EE 27, 48, 62, 252, 267, 272, 275, 290, 294–295, 298, 302–304, 408, 420, 422
 (also see Java 2 Enterprise Edition)
 application server 273, 320, 346
 applications 422
 container 296
 distributed environment 394
 specification 296
 Web service 302
JAAS 394, 411
 (also see Java Authentication and Authorization Service)
 API 394
 architecture 394
 login modules 394
Jakarta 387
 open source project 412
Japanese 396
Java 31, 45, 62, 95, 257, 261, 265–266, 271–272, 275, 282, 290, 294–295, 297, 304, 306–307, 310, 337, 350, 387, 405, 408, 410–411, 414, 425
 APIs 296
 classes for portlet classes 388
 client proxy 295
 Community Process 387
 development environment 295
 enterprise applications 408
 platform 296
 portlets 306, 422
 protocol 422
 Server Pages 305, 388
 server skeleton 295
 servlets 305
 Specification Request 62, 290
 standard security model 394
 Web Services Unleashed 295
Java 2 Enterprise Edition 273, 296, 320, 346
 (also see J2EE)
Java Authentication and Authorization Service 394
java.lang.String 305
JavaBean 412, 414, 422
 See Enterprise JavaBean
JavaDoc for Collaborative Components 412
JavaMail 357
JavaScript 306
JavaServer Pages 395
JAX-RPC 296, 302
 mapping 304–305
jaxrpcmapping.xml 297
JCA 26
JCP
 See Java Community Process
JD Edwards 12
JDBC 326, 345
JMS viii, 357
 processes 356
JMX 356
JNDI 280
Job Solicitation
 business/integration pattern 184
 use case 173, 175, 178, 181
JSPs 305–307, 389, 411, 414, 422
 (also see JavaServer Pages)
 alternate 310
 files 422
 pages 307, 412
 portlet 25
 syntax 306
 tag
 libraries 403
 references 413
 templates 395
JSR
 See Java Specification Request
JSR-101 290, 296, 302
JSR-109 290, 296, 304
 benefits 301
 deployment model 301

JSR-168 62, 306, 350

JSR-87 350

JXTA 356

K

key performance indicators 168–169

kiosks 168, 177, 194

K-map

(also see Knowledge Map)

Affinities interface 316

Building

log 318

Service 312, 315

Editor 312, 314

Metrics 313, 317

Profiles 312, 316

Spiders 312, 316

user interface 312–313

knowledge

management 311

worker 167

portals 42

Knowledge Map 312

Korean 397

L

Labor Scheduling functional subsystem 178, 181

language differences 76

layout 44

LDAP 96, 116, 253

(also see Lightweight Directory Access Protocol)

database 418

directory 276, 280, 308, 391, 399, 418, 422,

425

server 279

search parameters 319

server 418

service 423

user registry 164, 392

LDAP-compliant directories 390

legacy investment 18

Lightweight Directory Access Protocol 392, 422

Lightweight Third Party Authentication 392

lines of business 17

unified customer view 53

Linux 12, 27, 95, 257, 261, 265, 267–268, 271–272

live feeds 13

load balancing 47, 160, 163, 357

LOBs 397

(also see lines of business)

queries 21

local distribution across sites 160

locations 57

Log Files portlet 400

logical node view 69

Lookout ODS Service 143

Lotus 163, 228–229, 269

Collaboration Center 416

Directory Connector 423

People Finder portlet 423

Collaborative Components 403, 421

Discovery Services 404

Domino Data Service 404

Menu Service 404

People Service 404

QuickPlace Service 404

Team Workplace Service 404

Discovery Server 257, 311–312, 402, 421, 424–426

KDS API 411

knowledge maps 425

interface 403

Domino 228, 257, 320, 392, 403–404, 406, 411, 421

Directory 423

Domain Index 425

Enterprise Server and Notes 423

Java API 411

portlets 222

servers 379, 390

Tivoli Analyzer for 379

Web Access 417

Domino.Doc 312–314, 316, 318–319

Extended Search 390, 425

C++ and Java API 425

Instant Messaging 158, 222, 228, 232, 241, 243, 248, 251, 257, 316, 321, 386, 402–403, 411, 416, 421, 424–425

Contact List portlet 417

portlet 236, 240, 402, 423

server 163

Who Is Here portlet 417

Notes 225, 232, 386, 390, 411

calendar portlet 402

databases 312, 425

discussion portlets 402

document library portlet 402

e-mail portlets 402
Mail Portlet 236
TeamRoom portlets 402
to-do list portlet 402
Notes portlet 417
QuickPlace
 See Lotus Team Workplace
 See QuickPlace
Sametime
 See Lotus Instant Messaging
Software
 Collaborative Components
 API 411
 Team Workplace 243, 312–314, 316, 318–319,
 402–403, 411, 416, 421, 424–425
 portlet 423
 team room 386
Web
 Conferences 416
 Conferencing
 portlet 423
 Web Conferencin 232
 Web Conferencing 228
 Workplace Web Content Management 389
LTPA
 (also see Lightweight Third Party
 Authentication)
 cookie 392
 token 158, 164, 394
Lycos 43

M

machine sizing 194
macroflow 340
maintainability 17, 54, 95, 218
maintenance
 planning for 57
Manage
 and Schedule Work delta use case 214
 and Track Incident 99
 use case 88, 92, 102, 113
 Compensation and Benefits use case 174–175,
 178, 181
 Customer Accounts use case 195
 Supplier Accounts use case 195
Tasks
 business/integration pattern 184
 use case 173, 175, 178, 181

Users use case 214
manageability 95, 218
Managed
 Collaboration application pattern 184
Process
 application
 integration pattern 127, 137, 218
 pattern 220
Public
 and Private Process
 application pattern 125
Process
 application pattern 125
Management of External Data 325
MAPE 352
market
 segment 204
 share 50
marketing
 campaigns 12
 plan 204
marketplace mindshare 16, 52
matching the intended solution
 to the Portal composite pattern's use cases 91
measuring expectations 247
MED 325
 (also see Management of External Data)
MediaApps 388
meeting minutes 13
mega portals
 See public portals
member services 391
memos 13
mergers and acquisitions
 adaptability during 53
Message
 Broker 133, 135, 141
 logical node 143
 node 131
 Hub 138
MessageListener 307
metadata 95, 210, 213, 215, 220, 338, 344, 391,
405
 management 24
 and application development tools 341
 sharing 24
 standard interchange 339
 XML 343
microflow 340

Microsoft ix, 154, 288–291, 295, 306
Active
 Directory 392
 Server Pages
 See Active Server Pages
Excel 281
Exchange 425
 e-mail 312, 319
 productivity applications 386
 public folders 312, 314, 319
 spider 317–318
HTTP server 422
Index Server 390, 425
Internet
 Explorer 288, 310
 Information Server 288
Site Server 390, 425
SQL Server 390, 425
MIME types
 support 305
mission and vision statement 78
mobile
 devices 16, 406
 phones 406
Mobile portals 406
model office 26
Model-View-Controller 243
 (also see MVC)
 design 306
 pattern 410, 412
 pattern 387
Moreover (Portal partner) 388
MQ messaging 383
MQSeries
 See WebSphere MQ
MS
 Windows ix
MT
 (also see Machine Translation)
 engines 423
multiple
 delivery channels 17
 device support 75
multi-type device access 50
MVC 149
 (also see Model-View-Controller)
 design 306
 pattern 410
 pattern 243, 387
MyList portlet 417

N

NASSL
 (also see Network Accessible Service Specification Language)
Toolkit 291
national language support 414
nationality-based differences 76
navigation model 44
net meetings 14
Netegrity SiteMinder 392–394
Netscape
 Directory Server 392
 Enterprise Server 288
 HTTP server 422
 Navigator 288
network
 appliances 159
 structure 38
 topologies 376
 traffic distribution
 across multiple sites 160
 view 150
Network Accessible Service Specification Language 291
networked communities 60
newInstance method 302
NewsEdge 388
Newsgroup portlet 417
next-generation desktop 12, 63
NFRs 68, 72, 81, 94–95, 97, 151–152, 157, 163–164, 176, 218
 (also see Non-Functional Requirements)
 high-availability 161
 response-time 194
 security-related 163
 work product 97
node 356
 descriptions 150
 managers 356
Non-Functional Requirements 57
 work product 68, 84
non-repudiation 305
Notes
 See Lotus Notes
Notify Customer use case 115
Novell eDirectory 392

NTT DoCoMo iMode network 406

O

OASIS 293, 387

(also see Organization for the Advancement of Structured Information Standards)

Object IDs 419

object oriented design 308

Object Representation of Processes and Linking
delta use case 215

object-oriented software 28

object-relational

 data models 337

 queries 343

OCS channels

 See Open Content Syndication channels

ODBC 326, 345

ODS

 See Operational Data Store

ODW

(also see On Demand Workplace)

 collaboration services 196

 composite pattern 196

 generic use cases

 Access

 eHR System 195

 from PDA 195

 (partial) 195

 Help Desk 195

 Do Online Training 195

 Manage

 Customer Accounts 195

 Supplier Accounts 195

 View

 Analytic Data 195

 Corporate Information 195

 Syndicated Content 195

 Runtime pattern 196

 desktop 189

 portals 174

 Runtime pattern 196

 solution 167, 174

 asset 197

office documents 13

OGSA 269, 353, 355, 358

OGSI

 See Open Grid Services Infrastructure

OIDs

See Object IDs

OLAP 14

OLTP 340, 345

 systems 340

on demand

 business 4–6, 8–11

 characteristics 58

 definition 259

 capabilities 4, 6–7, 9, 11

 characteristics 208

 e-business 5

 environment 208

 era 3

 offering 10

 list 11

 partners 12

 offerings by industry 9

 operating environment 6–8, 10–11

 characteristics 59

 package 11

 portfolio 11

 world 4

On Demand Workplace 167, 195, 201, 229, 231

 solutions 196

online

 awareness 424

 meetings 403

Open

 Content Syndication channels 388

 Grid Services

 Architecture 269, 355

 Infrastructure 358

 Services Infrastructure 273

open

 framework 43

 on demand operating environment 59

 operational environment 208

 portal framework 15

 source

 communities 61

 software 288

 standards 4, 6, 8, 27, 59, 211, 257, 261,

 265–266, 268, 271–272, 275, 408, 422

operating system 65

Operational

 Data Store 143

 Model 72–73, 128, 142, 150, 156, 193

 definition 150

 development process 151

- with clustering 158
 - work product 72, 98, 157
 - operational**
 - domain 57
 - efficiency 209
 - models
 - design 95
 - operations portal 15
 - optimization 11
 - Oracle viii, 155, 324, 390–391, 425
 - data cartridges 337
 - Financials 405
 - order
 - issuing and configuration 247
 - tracking and completion 247
 - organizational efficiency 17, 53
 - outsourcing
 - model 10
 - overall business strategy 57
- P**
- page**
 - design 24
 - transformation services 47
 - page sequence tags 413
 - PARA**
 - (See parallel query processing techniques)
 - parallel query processing techniques 336
 - Participate in e-meeting use case 91
 - partitioning
 - horizontal 358
 - vertical 358
 - patterns** 4
 - Access Integration**
 - (See Access Integration pattern)
 - Application**
 - (See Application patterns)
 - box**
 - identifying 110
 - notation
 - IA (Find Agency Info) 106
 - Risk Management System 117
 - SS (Request Entry) 106
 - Business**
 - (See Business patterns)
 - business drivers** 121
 - Document Exchange 121
 - Exposed Application 121
 - Composite**
 - (See Composite patterns)
 - custom design**
 - (See custom design)
 - definition** 28
 - design** 28
 - Integration**
 - (See Integration patterns)
 - IT drivers** 122
 - mining** 196
 - Model-View-Controller** 387
 - Portal composite**
 - (See Portal composite patterns)
 - replicable** 107
 - Runtime**
 - (See Runtime patterns)
 - type standards** 28
 - Patterns for e-business** 5, 29–30, 34, 55, 67, 71, 74, 97, 104, 110, 131, 155, 165, 222, 414
 - definition 32
 - steps for using 38
 - PCM** 206
 - (also see Product Change Management)
 - PDAs** 45, 48, 307
 - (also see personal digital assistants)
 - PDM** 209, 214–215, 219, 225, 228–230, 232, 248, 250, 253
 - (also see Product Data Management)
 - products 210
 - systems 210–211
 - People**
 - Finder portlet** 402, 416
 - Management Solutions** 189
 - people**
 - awareness 403, 424
 - integration 4
 - links 423
 - PeopleSoft** 12, 169, 405
 - back-end systems 192
 - credential 194
 - HR**
 - applications and data 168
 - system 188
 - portlet 194
 - SCM system 169
 - Perform Account Maintenance Functions**
 - User Provisioning use case 115
 - performance** 57, 81
 - monitoring 24, 47

persistence 410
 services
 process-state 219
 workflow-state 219

persistent
 data 47
 TCP/IP connections 387

personal digital assistants 406

personalization 23, 35, 44, 50, 57, 75, 228, 384, 414
 rules 18–19, 420
 services 47

Personalize Home Page generic use case 214

personalized
 content 18
 information 40

Personalized Delivery application pattern 113

pervasive
 access 7
 computing devices 272
 devices 43

Pervasive Device Access
 access integration pattern 126, 195
 application pattern 138, 184
 runtime pattern 186

Pharmaceutical industry 248

phases 65

PhotoShop 389

physical
 hardware view 150
 model 157

Physical Operational Model 152

Pinnacor 388

PKI 96, 163
 (also see Basic authentication/authorization)

place awareness 403

Place Order use case 115

plan phase 57

platform independence 268

PLM 62, 220
 (also see Product Lifecycle Management)
 systems 211

policy
 editors and validators 354
 enforcement 278
 IBM view 353
 management 278

policy-based
 identity management 321

orchestration 11

policy-driven computing 353

Ponder 353

Population
 Crawl and Discover application pattern 113
 Single Step application pattern 113

pop-up windows 310

Port Authority Portal 143

port component reference 300

portability 95

Portal
 Content Organizer 416
 Credential Vault 194
 Toolkit 419

portal
 administration 419
 application 57
 architecture 23
 benefits 54, 75
 diagram 31
 limitations 55
 as distinguished from a Website 41

B2B 396
 See B2B portals

B2C
 See B2C portals

B2E
 See B2E portals

BI
 See BI portals

branding
 See branding

business
 drivers
 See business drivers portal
 See business portal

collaborative
 see collaborative portals

commerce
 See commerce portals

commerce-enabled
 See commerce-enabled portals

community of users 45

configuration
 file 387
 updates 419

consumer
 See consumer portal

credentials 194

CRM
 See CRM portal
customizing 44
definition 12, 39
development site 27
e-business
 See e-business portal
e-commerce
 See e-commerce portal
e-mail portlet 237
employee
 See employee portal
enabling
 a business to be responsive and variable 61
 integration 62
 on demand 27
enterprise 96
 information 45
 See enterprise portals
environment 47
executive overview presentation 26
federated 400
first generation 43
framework 45–46
functional aspects 43
fundamental characteristics 40
helping a business become resilient 62
horizontal
 see horizontal portals
implementation 17, 30, 53–54, 57
 requirements 52
infrastructure 47
Internet
 See Internet portal
intranet
 See intranet portals
JSP tags 413
leading to e-business on demand 61
navigation 419
 model
 See navigation model
new desktop and integration standard 387
operations 15
page 48
personalizing 44
procurement 15
providing focus 62
public Web 45
second generation 43

self-service
 See self-service portal
server 46, 48, 163–164, 309, 386–387
 advanced features 310
 always available 162
 credential vault 387
 page aggregation subsystem 407
 patch 154
services
 collaboration
 See collaboration services
 communication
 See communication services
 content management support
 See content management support
 event notification
 See event notification services
 page transformation
 See page transformation services
 personalization
 See personalization services
 search
 See search services
 security and access control
 See security and access control services
 user and group management
 See user and group management services

sites
 development and maintenance 45

skin
 See skin

solution 49
 best practices 55
 business drivers for choosing 51
 definition 50
 deriving value from 50
 guidelines 155
 IT drivers for choosing 53
 steps for building 57
 Web-based 39

technical aspects 45

theme
 See theme

third generation 43

toolkit 388

types 41

typical services provided 44

vertical

see vertical portals
when to consider using 49

Portal composite pattern 29–30, 54, 74–75, 77, 89, 94, 101, 110–111, 113–114, 116–117, 128, 174, 180–181, 212–213, 223, 228, 414

- Application pattern 76
 - (also see Application patterns)
 - input device characteristic 138
- associated use cases
 - Assess Risk 102
 - Authenticate and Authorize User 102
 - Author and Approve Bulletin 102
 - Establish Team and Environment 102
- Find
 - Agency Information 102
 - SMEs 102
- Generate
 - Alert 102
 - Incident 102
- Inspect Cargo 102
- Manage and Track Incident 102
- Publish Bulletins 102
- Register with Agency 102
- Request Entry 102
- Resolve Incident 102
- User Provisioning 102
- functional subsystems
 - Cross-Agency
 - Collaboration 102
 - Document Management 102
 - Information 102
 - Entry Management 102
 - Inspection 102
 - Risk Management 102
 - User Management 102
- generic use case 90, 93, 174
 - Access Enterprise Applications 91
 - Authenticate 91
 - Authorize 91
 - Create Content 91
 - Customize Home Page 91
 - Display Home Page 91
 - Participate in e-meeting 91
 - Publish Content 91
 - Register New User 91
 - Search for
 - Content 91
 - People 91
 - Synchronous Chat 91

Visit TeamRoom 91

Runtime pattern 54, 129–130, 225

- variant 226

Use Case Model 89–90

use cases 91

- Access Enterprise Application 175
- Authenticate 174
- Authorize 174
- Participate in e-meeting 175
- Review Corporate Information 175
- Search for Content 175
- Synchronous Chat 175

portalize

- existing Web application 26

portal-style SSO environment 113

Portlet

- API 307, 387, 415, 419
 - standardization 410
 - tags 410
- Development Guide 415

PortletContext.getService() method 387

PortletData object 308

PortletLog 307

portlets 24–25, 48, 221, 306

- 3Dcom 238
- administrative 398
- API tags 413
- application 388
 - builders 417
- cache
 - refreshing 410
- catalog 386, 389, 391, 402
- click-to-action 387, 413
- clipping 420
 - See clipping portlet
- collaboration 401–402
- collaborative 424
 - Discovery Server 402
 - Lotus QuickPlace 402
 - Lotus Team Workplace 402
 - Mail 402
 - My
 - iNotes 402
 - Notes Calendar 402
 - Notes
 - Discussion 402
 - TeamRoom 402
 - View 402
 - People Finder 402

Sametime Chat 402
Team Workspaces 402
To-do List 402
Web
 Conferences 402
 Page 402
container 45–48, 386, 409
cooperative 418
COTS
 See COTS portlet
creation 410
credential vault 410
custom
 See custom portlet
definition 386
Discovery Server 402
ePCM 237
guidelines for developing 410
iFrame
 See iFrame portlet
iNotes 417
internationalization 307
Internet Mail Box 417
iView
 See iView portlet
Java 422
JSP
 See JSP portlet
library 229
Log Files 400
Lotus
 Domino 222
 Notes
 calendar 402
 discussion 402
 document library 402
 e-mail 402
 Mail 236
 TeamRoom 402
 to-do list 402
 Web Conferencing 423
Lotus Team Workplace 402
markups 407
menus 419
modes 386
MyList 417
Newsgroup 417
Notes 417
parallel rendering 410

People Finder 402, 416, 423
PeopleSoft 194
Portal e-mail 237
Sametime 236, 240, 402, 423
 Contact List 417
 Who Is Here 417
security 410
Team Workspaces 402
third-party 409
title bar 45
transcoding technology 419
usage
 analyzing 410
Web
 clipping 399, 420
 Conference center 402
when
 other solutions are better 310
 they make the most sense 309
window 45
wizards 419
XML
 See XML portlet
XSL
 See XSL portlet
Portuguese 397
POS
 devices 194
 system 177
PQ70837 154
prefer-local optimization 357
pre-order feasibility studies 247
presentation
 layer 23
 server 188
Presentation Server 140–141
 logical node 143
presentation-rendering mechanism 50
proactive analysis components 376
problem management 364
process
 federation 228
 integration 107
 mapping 209
 orchestration 211, 228
procurement portal 15
Product
 Change Management 219
 Data Management 210

Lifecycle Management 62, 211
product
catalog 115
leadership 16, 52
mapping 38, 72, 154, 156–157, 212
definition 32
goal 154
selecting 38
version 65
Production Change Management 206
Project
Description 171
work product 68, 81, 84–85
Management 246
project manager 30
protocol binding 303
provisioning 11, 278
policy 278
public
portals 42
general
See general public portals
industrial
See industrial portals
workstations 177
Publish
and Manage User Content
delta use case 214
Bulletins 99
use case 88, 92, 102, 113
Content use case 91
publish and subscribe 14, 24

Q
QAD 12
QBIC
See Query by Image Content
Quality of Service 377
Query
by Image Content 391
Inventory use case 115
quick process initiation 365
QuickPlace
portlet 402
See Lotus QuickPlace

R
R5 Domain Index 390

Rational
Rose 257, 321
Unified Process 56
XDE 257
Modeler 273
Professional 273
real transactions, real e-business 59
realtime
computing 60
message feeds 337
Receive and Audit Inventory 184
use case 173, 175, 178, 181, 195
Recruitment functional subsystem 178, 181
Redbooks Web site 434
Contact us xv
redistributing work 159
reduced cost and operational efficiency 54
redundant
data 212
servers 163
reference
architectures 4–5, 67–69, 89
implementations 56
solutions 67
refresh option 310
Register
New User
use case 91
with Agency
use case 88, 91, 99, 102, 113
with Company use case 115
reliability 95
Remote
Document Search Service 143
Procedure Call 289
replicated machines 159
reporting 14
Request
Entry
use case 88, 92, 99, 102, 105, 108, 113,
140
for Change 206, 214, 244
request batching 159
request/response paradigm 46
requirements
functional 65
non-functional 65
resiliency 6–7, 10, 58, 208
Resolve Incident

use case 87, 99, 102, 113
response document 46
responsiveness 6, 58, 208
Retail industry 248
reusable assets
 large
 leveraging 65
reuse 17
reusing existing
 database connections 159
 IT assets 53
reverse proxy 135
Review
 and Guide Schedule
 business/integration pattern 184
 use case 173, 175, 178, 181
 and Perform Tasks
 business/integration pattern 184
 use case 173, 175, 178, 181
RFCs 206–207, 214, 244–245, 248
 (also see Requests for Change)
Rich Site Summary 388
Risk
 Assessment
 Diagram 145
 scenario 145
 System 143
 Management 246
 functional
 block 109
 subsystem 99, 102, 108
RIXML 339, 341
 (also see Research Information
 Markup Language)
RMI
 protocol 422
ROI 273
role-centric
 application paradigms 62–63
 approach 62
roles
 architect
 See architect
 associate 189, 194
 developers
 See developers
IT Architect
 See IT Architect
operational manager 189
program manager 236
 assistant 237, 239–240
project manager 240
 (also see project manager)
requestor 237
sales
 See sales role
user
 See user roles
Router application pattern 184, 187
routers 159
 fault isolation technology 376
RPC 289
 (also see Remote Procedure Call)
RSS
 (also see Rich Site Summary)
 content 389
 portlet 388
rules engine 43, 47, 50
rules-based filtering 421
Runtime pattern 30–31, 38, 54, 69–70, 72, 97,
129–130, 134–135, 155
 composite 196
 definition 32, 128
 IFB 196
 Logical Node view 128
 ODW 196
 selecting 38
runtime portal server 388
Runtime Server 150
RUP
 See Rational Unified Process

S

sales 42
 role 29
 support 15
Sametime
 See Lotus Instant Messaging
SAP viii, 26, 239, 253, 339, 343, 405
 portlet builder 405
 R/3 viii, 225, 232
scalability 17, 45, 54, 62, 95, 211, 218
scaling
 horizontal 358
techniques
 aggregate user data 160
 batch requests 160

cache data and requests 160
Content Delivery Services 160
create machine cluster 160
local distribution across sites 160
minimize connections 160
segment the workload 160
use
 a special machine 160
 faster machine 160
schema mapping 339
SCL 291
 (also see SOAP Contract Language)
SCM
 See supply chain management
SDL 291
 (also see Service Description Language)
Search
 for Content
 use case 91
 for People
 use case 91
search
 and navigation 24
 and taxonomy 16
 services 47
security 11, 16–17, 57, 81
 and access control services 47
 management tools 379
 model
 enterprise-level 211
 patch deployment 376, 382
 proxy 140, 196
 requirements 194
 servers 163
segmented workloads 159
SEI 302, 304
 (also see Service Endpoint Interface)
self care 392
self-configuring 363, 370
 components 349
 environment 372
 IT infrastructure
 implementation levels 371
self-healing 363, 370
 and availability management
 implementation levels 374
capabilities 7
components 349
IT infrastructure 373
self-management technologies
 structure 366
self-managing systems
 components 348
self-optimizing 363, 370
 actions 378–379
 components 349
IT infrastructure
 optimizing workloads and transaction performance across multiple resources 377
self-protecting 363, 370
 components 349
 IT infrastructure implementation levels 380
self-service 42, 44
 human resource systems 167
 portal 15
Self-Service business pattern 33, 75, 103–105, 110, 113
self-syndication 388
Sell-Side Hub composite pattern 35
Send Order to Warehouse use case 115
server
 programming model 302
 reconfiguration 376, 382
 virtualization 270
Server Management Services 12
Service
 Description Language 291
 Endpoint Interface 302
 Web 282, 287–288
service implementation bean 302
Servlet API 307
servlet-based implementation bean 302–303
servlets 25–26, 48, 275, 305, 386, 408, 414, 422
 container 409
 containers 296
 engine 358
session EJB lifecycle 303
setSessionContext method 302
setting the agenda for the industry 8
SGML 272
 (also see Standard Generalized Markup Language)
Siebel 405
Simple
 Mail Transport Protocol 290
 Object Access Protocol 62, 282
Simple Object Access Protocol 400

single
authentication point for user interactions 140
point
of access 12, 18, 20, 39, 41, 45, 62–63, 167, 425
of integration 43
of personalized interaction 231
sign-on 16–17, 35, 40, 53–54, 75–76, 89, 177, 222, 228, 381, 384, 391, 393, 403, 406
user interface 43
versus multi-tier Portal server design 149
site
analytics 16, 384
design 12
skins 44, 308–309, 407, 414, 418
SLAs
compliance monitors 355
effectively meeting 378
small and medium business 27
SMB
See small and medium business
SMS messaging 113
SMTP 290
(also see Simple Mail Transport Protocol)
SNMP traps 376
SOAP 282, 288, 296, 339, 345, 400
(also see Simple Object Access Protocol)
1.1
specification 290
standard 289
1.2 specification 290
Contract Language 291
Toolkit 290–291
Web service 293
with Attachments 295
SOAP/HTTP 289, 291
SOAP4J 291
software
components 96
nodes 71
structure and modularity 80
distribution 57, 81
placement 81
product requirements 95
server 96
six-stage process for deployment 359
view 150
Solaris viii
solution 39

design
logical 97
development methodology
Step 1 - Develop solution requirements 68, 84
Step 1A - List users/existing systems and Functional Requirements 68, 85, 171
Step 1B - Derive actors and use cases 68, 86
Step 1C - Inventory large reusable assets 68, 89, 174
Step 1D - Match solution use cases with generic use cases 68, 90, 174
Step 1E - Identify delta use cases 68, 93, 175
Step 1F - Identify and document NFRs 68, 94, 176
Step 1G - Validate requirements with customer 68, 97
Step 2 - Develop the logical design 69, 97
Step 2A - Derive initial AOD 69, 98, 177
Step 2B - Seed the AOD with large reusable assets 69, 101, 179
Step 2B1 - Validate the AOD 69, 111
Step 2B2 - Capture chosen patterns into Architecture Decisions 69, 115
Step 2C - Identify and apply patterns for each AOD delta requirement 69, 116, 182
Step 2C1 - Identify delta requirements that need to be addressed in the AOD 69, 116
Step 2C2 - Identify and apply Business and Integration patterns for the delta requirements 69, 118
Step 2C3 - Identify and apply Application patterns for delta Business and Integration patterns 69, 118
Step 2C4 - Capture chosen patterns into Architecture Decisions 69, 127
Step 2D - Transition the AOD from subsystem view to logical node view 69, 128, 185
Step 2D1 - Seed the AOD (Logical Nodes) 69, 128
Step 2D2 - Identify and apply Runtime patterns for the delta Application patterns 70, 130
Step 2D3 - Integrate Runtime pattern for the Application patterns into the AOD 70, 133
Step 2D4 - Capture chosen patterns into Ar-

chitecture Decisions 70, 135
Step 2E - Review and apply logical design guidelines to the AOD (Logical Nodes) 70, 139
Step 2F - Conduct technical walkthroughs to ensure that functional requirements are met 70, 140, 188
Step 3 - Develop the Physical Design 71, 142
Step 3A - Build the Component Model 71, 142, 190
Step 3B - Apply design guidelines (application level) from matched large reusable assets 71, 149, 190
Step 3C - Conduct technical walkthrough of component interaction 71, 150, 190
Step 3D - Develop the Operational Model 72, 150, 192
Step 3E - Seed the Operational Model 153, 192
Step 3F - Address operational NFRs in the Operational Model 72, 157, 194
Step 3G - Conduct technical walkthroughs to ensure that NFRs are met 72, 161, 194
work AOD
 associated use cases
 Generate Incident 99
work product
 115
 Architecture Decisions 69–70, 81, 97
 Architecture Overview Diagram 69, 81, 97–98
 (Subsystem) 103–105, 108–111, 113–114, 116–118
 Also see AOD
 Component Model 71, 98
 Relationship Diagram 144
 Current System Context 85
 Envisioned Goals and Issues 80–81
 Functional Requirements 84
 High Level Project Description 68, 85
 Non-Functional Requirements 68, 84, 97
 Operational Model 72, 98, 157
 Project Description 81, 84–85
 System Context Diagram 68, 81, 83–85
 Use Case Model 68, 85–86, 89, 94
development methodology activity
Demonstrate Business Benefits and Capabilities 68
Develop
 Customer Interest 67
 Plan Linked to Customer's Business Initiatives 67
 Solution with Customer 68
Establish Buying Vision with the Customer 67
Evaluate Customer's Environment 67
Monitor Solution Implementation and Ensure That Expectations Are Met 73
Qualify Opportunity 68
Refine Solution, Resolve Concerns, Close Sale 73
development methodology phase
Execute 66–67, 73, 80, 169
 (develop solution requirements) 170
 (develop the logical design) 177
 (develop the physical design) 190
Implement 65–66, 73, 165
Plan 66–67, 77, 169
development methodology task
Analyze Current IT Infrastructure 67
Assess
 Business Impact 73
 Initial Viability 68
Define Business Context 67
Describe Current Organization 67
Develop
 High-level
 Component Model 71
 project description and System Context Diagram 81
 Operational Model 72
 Solution Architecture 69
Document
 Envisioned goals and issues 77
 IT Standards 67
Ensure Client Commitment 73
Evaluate
 Integrated Solution 73
 Success 73
Gain Sponsorship 68
Harvest Assets 73
Identify Business Issues and Goals 67
Monitor Pilot 73
Obtain or Develop Business Roadmap 67
Outline Solution Requirements 68

Refine Viability Assessment 73
steps in designing 81
use cases that match Portal composite pattern
generic use cases
 Author and Approve Bulletin 92
 Authorize and Authenticate User 91
 Establish Team and Environment 91
 Find SMEs 92
 Manage and Track Incident
 and Resolve Incident 92
 Publish Bulletin 92
 Register with Agency 91
 Request Entry 92
Spanish 396, 423
Specified Operational Model 152
SQL 327, 333, 335, 340, 343
 dialect 323
 Server 405
 statements 295
SQLMM 344
SQLX 337, 340–341, 343, 345
SSO 164, 177, 194, 393
 (also see single sign-on)
 account information 149
 best practices 149
 guidelines 155
 interface 149
 usage guidelines 149
standalone availability 357
Standard Generalized Markup Language 272
standards 24
 bodies 61
Starburst 336
Start Now
 Collaboration Solutions 12
 Infrastructure Management Solutions 12
static stub 304
Stellant 389
stock management system 169
StockQuoteService.java 294
StockQuoteServiceSoapBindingStub.java 294
storage virtualization 270
Store and Retrieve application pattern 113, 184
storyboards 97
stovepipes 83, 119
 applications 107
structured data 44
Struts 310
 framework 387
portlet framework 412
 page sequences extension 412
Website 412
WML tags 413
struts-config.xml 412
subsystem view 69
Sun
 Microsystems viii, 155
 ONE 424
supply
 chain 7, 14
 management 22, 41, 384
 performance management 14
 systems 168
support
 alerting 14
 for mobile devices 384
Sybase 322, 390, 425
Synchronous
 Chat use case 91
 Collaboration delta use case 214
syndicated content 44
 feeds 386, 409
Sysplex Timer and coupling facilities 376
System
 Context Diagram 171
 for e-government 84
 work product 68, 81, 83–85
 Sequence Diagram 150
 Topology Diagram 150
system
 interoperability 59
 management 57, 81
 organization 57, 81
 requirements
 documenting 66
systems architecture diagram 31

T

TAI

(also see Trust Association Interceptor)
custom modules 393
See Trusted Association Interceptor

TAM 164

(also see Tivoli Access Manager)
V4.1 154

Tandem 336

target audience 51

targeted users 50

Task Management

- functional subsystem 178, 181
- functionality 174
- system 168

tasks 65–66

taxonomy 16

- See *Knowledge Map*

TCO 229

- (also see *total cost of ownership*)

TCP/IP networks 376

Team Workplace 158, 163, 251

Team Workspaces portlet 402

Teamcenter Enterprise viii, 231–232, 253

TeamRooms 113

technical walkthrough 70–73, 140, 161

techniques for achieving scalability 159

technology domain 30

Technology Model 293

text mining 13

themes 44, 308–309, 407, 414, 418

thick client 277

third-party

- portlets 409
- reverse proxy servers 393

time to value 273

time-to-market 17, 20, 53

- reducing 208

Tivoli 269

- Access Manager** 164, 194, 368, 381, 392–394
 - for e-business 426
- Analyzer for Lotus Domino** 379
- Business Systems Manager** 369, 376, 379
- Configuration Manager** 368–369, 372

Enterprise

- Console** 375, 378–379
- Data Warehouse** 378

Identity

- Director** 369
- Manager** 257, 277, 321, 372, 382
 - Application Interface module** 278
 - Applications subsystem** 277
 - Core Services subsystem** 278–279
 - DSML connectivity** 280
 - IBM Directory Integrator**
 - connectivity 281
 - JNDI API** 280
 - LDAP directory** 280
 - logical component architecture** 276

Platform subsystem 277–278

Resource connectivity 279

server 278, 280

Web User Interface subsystem 277–278

Monitoring 368

- for Applications 376
- for Databases 376
- for Middleware 376
- for Transaction Performance 369, 379

NetView 376

Privacy Manager for e-business 382

Risk Manager 376, 382

servers and products 163

Service Level Advisor 369, 378

Storage

- Manager** 368, 373, 379, 381
- Resource Manager** 369, 377

Switch Analyzer 376

Systems Automation for S/390 369, 376

Web Site Analyzer 397, 400, 424

- portal usage reports 398, 401

Workload Scheduler for Applications 378

tModel 293

- (also see *Technology Model*)

topology 57

total cost of ownership 17–18, 53, 230

trading

- exchanges** 14
- partner agreements** 125

Training functional subsystem 178, 181

transactional

- and process efficiency 17, 52
- services**

See *self-service*

Transactional application integration pattern 127, 137

transcoding 414

- technology 407

transparent business process 209

Transportation industry 201–203

- key business drivers 208

Trigger/Control Application Level Workflows

- delta use case** 215

Trust Association Interceptor 406

Trusted Association Interceptor 164

trusted network 164, 196

turnover 169

U

U2B 221

(also see User-to-Business)

pattern 104

Topology 4 222

U2D pattern

See User-to-Data pattern

U2U pattern

See User-to-User pattern

UDDI 282, 296

(also see Universal Description, Discovery, and Integration)

Business Registry 289, 293, 295

Network 291

directory 399–400

entities 293

specification 293

UGS PLM Solutions viii

UML 56, 321

(also see Unified Modeling Language)

unauthorized access and use 379

unified logging and tracing 360

Unified Modeling Language 66, 321

Universal Description

Discovery and Integration 282

Universal Description and Discovery Interface 400

UNIX 267

unmatched (delta) use cases 175

unstructured data 44, 113

Update Personnel Record use case 174–175, 178,

181, 188–189, 195

usability 95

use case 81

generic 89

logical grouping

See functional blocks

methodology 56

Use Case Model 68, 87, 172

work product 68, 85–86, 89, 94

User

Dictionary Manager tools 423

Management

business/integration pattern 184

functional

block 99

subsystem 99, 102, 177, 181

node 130

use case 174–175, 177, 181

Provisioning 99, 115, 140

use case 88, 99, 102, 113

Registration use case 214

user

and group management services 47

authentication and authorization 47

communities 14

data aggregation 159

database 45

interface

formatting and display 50

integration 305

loyalty 45

management 228, 406

preferences 12

profile 47

registration 228

roles 18–19

satisfaction 57

user-friendly URLs 419

users 33, 104

User-to-Business pattern 104

(also see Self-Service business pattern)

User-to-Data pattern

See Information Aggregation business pattern

user-to-user

interaction examples 222

User-to-User pattern 33

(also see Collaboration business pattern)

utility computing 8, 10, 257, 261, 265, 270–271

utility-like computing 8, 10, 271

utilization costs 260

V

Value Chain Management 62

value chain processes

managing 274

value-added resellers 22

variability 58, 208

variable cost structures 6

VCM

See Value Chain Management

Verity 389

vertical

portals 16

(also see industrial portals)

definition 15

Viability Assessment 73

View

Analytic Data use case 195
and Send e-Mail use case 214
Corporate Information use case 195
Syndicated Content use case 195

Vignette 388

Virtual

- Private Network 133
- Product Model 249

virtual

- collaborative organizations 355
- computer 59
- data warehouse 332
- database 323
- enterprise 15
- office 43

virtualization 6–7, 11, 271

virtualized

- networks 321
- on demand operating environment 59
- operational environment 208

viruses 379

Visit TeamRoom use case 91

Visual Basic 290, 310

VPM

- (also see Virtual Product Model and ENOVIA VPM)

VPN 141

- (also see Virtual Private Network)

end point 131, 135

W

W3C 288–291, 353

- (also see World Wide Web Consortium)

Web

- Accessibility Initiative 396
- Services Description Working Group 291

walkthrough 150, 176

- description 151
- SSO 194
- steps 162
- Update Personnel Records use case 189

WAP 282, 406

- (also also see Wireless Application Protocol)
- phone 286
- WML 19

WAR

- files 399
- (also see Web archive files)

WBI 243

Web

- access
 - client 423
- application
 - server 423
- archive files 388
- clients 46
- clipping 414
 - portlet 399, 420
- conference 424
- Conference center portlet 402
- content 386
 - dynamic 408
 - management 388, 420
 - products 389
 - system 388–389
 - publishing 384, 389
- Content Publisher 389
- crawler 391
- delivery channel 61
- documents 13
- enablement
 - front-end 96
- infrastructure
 - leveraging 63
- interface
 - business content 13
- pages 386
- portals that aggregate data
 - from disparate systems 80
- publishing 414
- server 48, 196, 272, 358, 360, 388
- Service Description Language 282
- services 8, 27, 253, 257, 261, 265, 267–269, 271–273, 282, 284–285, 287, 289, 291, 293, 301, 304, 308, 320, 345–346, 358, 384–386, 408–409, 422
 - Activity 290
 - assembler role 299
 - broker 309
 - building blocks 295
 - client 326
 - composition 291
 - container 296
 - definition 400
 - deployer role 301
 - description documents 344
 - developer role 298

development tools 295
for J2EE 296
 development flow 297
 platform APIs 296
for J2EE product provider 300
guidelines 149, 155
infrastructure 296
Interoperability Organization 295
mechanisms for invoking 304
provider 309
requester 309
standards 288, 290
stubs 300
Toolkit 293
tools 294
Single Sign-on application pattern 113
test client 295
top 15, 40
 (also see portal)
User Interface 277
Web Archive 399
Web Service Description Language 400
Web-enabled wireless phones 307
WebSEAL 158, 163–164, 194, 392
webservices.xml 297
Website services 24
WebSphere 159, 269
 Application
 Server 257, 273, 282, 290, 295, 320, 346,
 355–356, 358, 383, 419–420, 422, 424
 Advanced Edition 408
 InfoCenter 393
 security 406
 Single Server test environment 275
Application Server 392
Business
 Integration 249, 274
 Server 257, 273, 320, 346
 Integrator 384
Business Integratio 243
Commerce Server 19
 Integration with WebSphere Portal 406
Edge Server 159, 427
Interchange Server 249
 EAI platform 228
Member Services 418
MQ 228, 243, 249, 274–275
 Workflow 228
Personalization
campaign management tools 397
collaborative filtering capabilities 397
recommendation engine 397
Server 397
Platform
 IBM WebSphere Portal 383
 Reach and User Experience 384
Portal 16, 19, 23, 25–26, 229, 232, 248, 310,
404
 APIs 414
 Application
 Integration 192
 architecture 384–385
 393–397, 401, 406
 administration 392
 advanced personalization 385
 application connectivity 385
 authentication 392
 proxy 392
 business process integration 385
 cascading page layouts 385
 data synchronization 385
 delegated administration 385
 document indexer 389
 federated search 390
 intelligent notification 385
 knowledge management 385
 member services 391
 authentication 391
 authorization 391
 single sign-on 391
 user and group subsystem 391
 offline browsing 385
 pervasive computing functions 385
 portal federation 385
 search portlet 389
 third-party
 authentication server 392
 Web crawler 389
Catalog 27
developer
 basic skills 408
 information roadmap 408
developing portlets 409
Enable 27, 414
Experience 27, 389
Express 12, 27
Extend 27, 402, 414, 423
framework 420

installation/configuration 409
integration
kits 389
with WebSphere Commerce Server 406
integration with ePCM 243
Member Manager 418
migration 409
Server 16, 19, 62, 228, 307–309, 397, 411
portlets 306
Toolkit 414–415, 421
what's new with Version 5.0 416
Zone 27
Portlet API 306
portlet catalog 415
resource framework interfaces classes 397
SDK for Web Services 308
software programming model 427
Studio 257, 273, 293, 295–296, 307, 320, 346
Application Developer 274–275, 388, 400, 421
desktop 294
Site Developer 421–422
Workbench 415
Translation Server 396, 422
WebSphere Business Integration 228
framework 228, 237, 248
WebSphere Portal Application Integration 192, 194, 404
WebSphere Portal Content Publishing 420–421
WebSphere Portal Server 164
whiteboarding 14, 44, 214, 228, 251, 253–254, 403
WindowListener 307
Windows
2000 ix
Active Directory 424
NT ix
platforms 306
WIP
See Work in Progress
Wireless
Application Protocol 282
Markup Language 282
wireless
gateway node 186
WLM 357
(also see workload management)
controller 357
routing algorithm 357
WML 282, 310, 406
(also see Wireless Markup Language)
Word (Microsoft) 389
Work
Notifications and e-Mails delta use case 215
Work in Progress 244
work products 65–66
dependencies 74
primary 73
workflow
federator 228
systems 337, 339
Workflow Monitoring delta use case 215
workload
balancing 378
distribution server 159
management 357, 377
World Wide Web Consortium 272, 288
WS
(also see Web services)
Policy
enablement 355
group 353
Security 295
WSDL 282, 289–290, 293, 295–297, 300, 304, 345, 400
(also see Web Services Description Language)
1.1 specification 291
1.2 291
mapping rules 302
port 304
Toolkit 293
WS-I.org 295
(also see Web Services Interoperability Organization)

X

X-Agency
Response Team 116
team 87–88
XDE 273
(also see eXtended Development Experience)
XHTML 46
XMethods 290
XML 19, 24, 253, 272, 274–275, 282, 293–294, 296–297, 304, 312, 318, 320, 334, 346, 356, 387, 390, 400, 408, 414, 419, 426
(also see Extensible Markup Language)
access 419

based protocol 279
data
 model 343, 345
 retrieval 340
development tool 295
document life cycle 344
documents 275, 337, 422
DTD 324
encryption 305
extensions to SQL 337
metadata 343–344
portlet 25
Protocol Working Group 290
query language 340
Registry
 WebDav-compliant 344
repositories 289, 291, 339, 343
request body 291
schema 289, 291
 definition 280
 documents 344
semi-structured documents 342
SOAP messages 295
 transformation 386
XML-based deployment descriptors 302
Xperanto 320, 346
 (also see DB2 Information Integrator)
XQuery 340–341, 343, 345
XSD 289, 295
 (also see XML schema)
XSL portlet 25

Y

Yahoo 42–43
YellowBrix 388

Z

zSeries 27

IBM



Redbooks

Architecting Portal Solutions

(1.0" spine)
0.875" <-> 1.498"
460 <-> 788 pages

Architecting Portal Solutions



Positioning portals as an integral part of e-business on demand

Using IBM best practices to architect portal solutions

Extending portal solutions with IBM Business Partners

A redbook focused on architecting and building WebSphere Portal Server-based Dynamic Workplaces. This redbook is aimed at addressing the needs of SWG Architects, Business Partners, and Customers' needs for building skills in architecting solutions by identifying themes that can be applied across multiple SWG Industry Solutions. This includes using e-business infrastructure solutions maps that apply across multiple industries (e.g., the on demand workplace solution, the e-business integration stack from SWG that provides capabilities for integrating the enterprise based on both business and technology drivers, and the IBM patterns for e-business from SWG that provide the basis for understanding the best practices for supporting e-business integration. This redbook provides real examples of industry-specific Dynamic Workplace solutions from leading ISVs and Systems Integrators working within IBM's on demand strategy. It also includes information on how to use IBM software capabilities to fulfill customer requirements using WebSphere, Lotus, Rational, Tivoli, and Data Management products, how portals are used in various industries, and how this investment can lead a customer towards becoming an e-business on demand.

About the title: The use of "to architect" and "architecting" as verbs are emerging in the field to address the limitations imposed by the verbs "design" and "build" to capture what architects do. In this redbook, the word "architecting" is used to describe all of the activities that architects perform to develop a solution, including understanding the customers' environment, understanding and defining the requirements, analyzing the system, designing the solution, and managing the solution development process.

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