



# IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition)

By Chris Britton, Peter Bye

[Download now](#)

[Read Online](#) 

## IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) By Chris Britton, Peter Bye

The challenges of designing, building, and maintaining large-scale, distributed enterprise systems are truly daunting. Written by and for IT professionals, *IT Architectures and Middleware, Second Edition*, will help you rise above the conflicts of new business objectives, new technologies, and vendor wars, allowing you to think clearly and productively about the particular challenges you face.

This book focuses on the essential principles and priorities of system design and emphasizes the new requirements emerging from the rise of e-commerce and distributed, integrated systems. It offers a concise overview of middleware technology alternatives and distributed systems. Numerous increasingly complex examples are incorporated throughout, and the book concludes with some short case studies.

Topics covered include:

- Middleware technology review
- Key principles of distributed systems: resiliency, performance and scalability, security, and systems management
- Information access requirements and data consistency
- Application integration design
- Recasting existing applications as services

In this new edition, with updates throughout, coverage has been expanded to include:

- Service-oriented architecture concepts
- Web services and .NET technology
- A more structured approach to system integration design

 [Download IT Architectures and Middleware: Strategies for Bu ...pdf](#)

 [Read Online IT Architectures and Middleware: Strategies for ...pdf](#)

# IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition)

By Chris Britton, Peter Bye

**IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition)** By Chris Britton, Peter Bye

The challenges of designing, building, and maintaining large-scale, distributed enterprise systems are truly daunting. Written by and for IT professionals, ***IT Architectures and Middleware, Second Edition***, will help you rise above the conflicts of new business objectives, new technologies, and vendor wars, allowing you to think clearly and productively about the particular challenges you face.

This book focuses on the essential principles and priorities of system design and emphasizes the new requirements emerging from the rise of e-commerce and distributed, integrated systems. It offers a concise overview of middleware technology alternatives and distributed systems. Numerous increasingly complex examples are incorporated throughout, and the book concludes with some short case studies.

Topics covered include:

- Middleware technology review
- Key principles of distributed systems: resiliency, performance and scalability, security, and systems management
- Information access requirements and data consistency
- Application integration design
- Recasting existing applications as services

In this new edition, with updates throughout, coverage has been expanded to include:

- Service-oriented architecture concepts
- Web services and .NET technology
- A more structured approach to system integration design

**IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition)** By Chris Britton, Peter Bye Bibliography

- Sales Rank: #982272 in Books
- Published on: 2004-06-03
- Released on: 2004-05-24
- Original language: English
- Number of items: 1
- Dimensions: 9.00" h x .90" w x 6.20" l, 1.06 pounds
- Binding: Paperback

- 368 pages



[Download IT Architectures and Middleware: Strategies for Bu ...pdf](#)



[Read Online IT Architectures and Middleware: Strategies for ...pdf](#)

## Download and Read Free Online IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) By Chris Britton, Peter Bye

---

### Editorial Review

#### From the Back Cover

The challenges of designing, building, and maintaining large-scale, distributed enterprise systems are truly daunting. Written by and for IT professionals, "IT Architectures and Middleware, Second Edition," will help you rise above the conflicts of new business objectives, new technologies, and vendor wars, allowing you to think clearly and productively about the particular challenges you face.

This book focuses on the essential principles and priorities of system design and emphasizes the new requirements emerging from the rise of e-commerce and distributed, integrated systems. It offers a concise overview of middleware technology alternatives and distributed systems. Numerous increasingly complex examples are incorporated throughout, and the book concludes with some short case studies.

Topics covered include: Middleware technology reviewKey principles of distributed systems: resiliency, performance and scalability, security, and systems managementInformation access requirements and data consistencyApplication integrationRecasting existing applications as services

In this new edition, with updates throughout, coverage has been expanded to include: Service-oriented architecture conceptsWeb services and .NET technologyA more structured approach to system integration design

#### About the Author

**Chris Britton** is an independent consultant, specializing in IT architecture. He has worked in IT for the last twenty-seven years doing a variety of jobs—programming, technical support, system software design, program management, technical consultancy, and even marketing. More recently he has been spending his time developing an IT modeling tool.

**Peter Bye** has had a long career in IT as a programmer, analyst, and project manager, focusing on telecommunications, transaction processing, and distributed systems. Currently a system architect for Unisys Corporation, his experience includes work in software development centers and field projects around the world. Peter is a contributor to international standards in systems management, the author of a number of papers on middleware and other IT topics, and a frequent speaker at conferences and other events.

Excerpt. © Reprinted by permission. All rights reserved.

All large organizations have complex, heterogeneous IT systems. All of them need to integrate their applications to support faster, more accurate business processes and to provide meaningful, consistent management information. All organizations are struggling to achieve this.

One reason for their struggle is that they are caught in the crossfire of an IT vendor war. In one corner is Microsoft Corporation, whose strength is its consistent technical strategy based on the .NET set of software technologies and the Windows operating system. In the other corner, ranged against Microsoft, is a group

that includes IBM Corporation, Sun Microsystems, Oracle Corporation, and BEA Systems. This group is focusing its resources on the Java environment. This is a battle about who will rule over middleware technology, a battle about how to implement distributed systems. Given the importance of the subject matter, it is a battle for the hearts and souls of IT for the next decade. Why? Because all large organizations have complex, heterogeneous IT systems that need to be brought together.

Vendor wars are only part of the problem. Scratch the surface of a large IT department and you will see many camps, for example, workstation/departmental server "decentralizers" and mainframe "centralizers." Look from another angle and you will see two kinds of people—"techies" and "modelers." A techy will start a project by deciding what platform and software to use and will eventually get around to the boring bit, which is writing application code. A modeler will design the application with a modeling tool, generate a few programs and a database, and will eventually confront the (to him or her) trivial question of what platform it will run on. Modeling to a techy seems abstract and disconnected from reality. Technical issues to a modeler are tedious, and surely, soon we will be able to generate the application from the model at the press of a button, won't we? One key to developing large distributed systems is to bring these people together.

Computer professionals generally are comfortable with developing applications on a single platform to a well-defined set of requirements. The reason is that the technology is well understood; the modelers know that what they design can be implemented, and the techies know they can make it work. Large distributed systems are not like that. A system designed without consideration for the distributed implementation will not work. Even worse, it will only become apparent that it doesn't work when scaling up to production capacity starts. To add to our woes, we are now considering integrating multiple systems, each of which was a challenge to develop in the first place, and each of which is changing at a different speed, driven ever faster by the business. The notion of a "well-defined set of requirements" is not realistic; requirements will always be changing.

It is our contention that modelers need to know something about technology, techies need to know something about modeling, and while we are about it, vendors, commentators, consultants, academics, and marketers need to know why their "solutions" aren't quite the panaceas they claim.

This book is about IT architecture. IT architecture provides a framework for discussing implementation design, and it is in these discussions where techies and modelers should meet. Anyone whose roles and responsibilities include IT architect should know everything in this book. (Note we said "know," not "agree with.")

Although IT architects are an important audience for this book, we have tried to write a book for IT management. We have assumed that the IT managers in our readership come from an IT background not a business background; this book is not an introduction to IT. So why do IT managers need a book about IT architecture? It is because here so many of their concerns come together—application flexibility, information quality, resiliency, scalability, and so on. One of our aims is to give IT management the knowledge to call the IT architects to account.

This book gives an overview of the whole subject of building and running large distributed systems. It is a deliberate attempt to step above the detail and the in-fighting to examine what is important, what isn't important, and what we need to do differently from the way we did it 10 years ago. Our contention is that the difference between then and now is much more than simply that there are new tools to play with. Building integrated systems is substantially different from building standalone applications, and it affects everything we do in IT.

A major theme of this book is enterprise computing. In the list of terms abused by the industry, "enterprise

computing" has to be somewhere near the top. This book takes the view that enterprise computing is about being able to build systems that support the whole enterprise, which in large organizations means many thousands of users. It's obvious that systems that support thousands of users must have resiliency, scalability, security, and manageability as major concerns. The enterprise computing mentality is not prepared to compromise on these objectives. An old mainframe application written in COBOL that gives you resiliency, scalability, security, and manageability is far superior to any implementation that does not.

This is not to say that you cannot build enterprise-capable applications with modern tools, such as Web services. But to succeed you must understand the principles of building large, resilient systems. The principles that served us well for standalone applications do not all apply for distributed systems and vice versa.

Many people in IT find discussions of principles too abstract and dry, so we have tried to enliven the presentation with many examples.

Many organizations today are trying to avoid all these issues by buying third party application packages. This is partially successful. When you buy a package, you buy an IT architecture, albeit only in the context of the package's functionality. If you buy many packages, it is likely that you must lash them together somehow, and for this you need an IT architect. If the packages are from different vendors, integration is a challenge. In this book, we give you the principles that should help in this task, but we have chosen not to address the challenge directly. Because there are so many packages, to do the subject justice would require another book.

This book is not for everyone. You will find this book short on product detail. It does not tell you anything about installation, there are no proper coding examples, there is no survey of products, and there is little in the way of product comparisons. We make no apologies for any of these omissions. There are many books on coding, and product details change so fast the best place for comparisons is on the Internet. This book does not teach application design. There are many books for this as well. But we hope application designers will read this book because the discussion on the principles for building enterprise systems is vital for them also. Finally, this book is not an academic book. It contains little mathematics except for the back-of-the-envelope calculations to illustrate a few points. We offer a practical, wide-ranging discussion for IT professionals to help them understand what is going on so they can pick out the real issues from the imaginary issues and start building complex distributed systems with confidence.

## **Changes in the second edition**

Changes that led to the second edition fall into three categories: new author, new technology, new approaches to design.

First, authorship has now become a team—Peter Bye has joined Chris. Peter brings a long-standing expertise in integration, networking, systems management, and all matters concerned with designing an IT architecture. Although Peter has concentrated on adding to the technical content in the first part of the book, all parts are mulled over by both of us and authorship is shared. Compare the new text with the old and you will notice small changes scattered everywhere. As befits Peter's additional expertise, where there was one chapter on systems management and security, there are now two.

Second, the book has been updated to take into account new technology, in particular Web services, and consequently new ways of thinking about IT architecture, in particular loosely coupled architectures.

Chris was in the final stages of writing the first edition of the book when Microsoft announced .NET and he had no time to digest the merits of the announcement and incorporate it into the text. When Web services

moved toward center stage, it became clear that a fundamental driver was a desire for a more loosely coupled architecture or, more specifically, a loosely coupled integration of tightly coupled archipelagos of system. The integrated applications architecture described in the first edition of this book was a quintessentially tightly coupled approach. When many people said that they could never reach enough agreement across an organization to impose a tightly coupled architecture, we could see some truth to these complaints. But loosely coupled integration is impossible without some common standards, and Web services are now providing them.

However, "loosely coupled" is a fuzzy term, easily bandied about by salespeople and consultants, and you can't just apply Web services pixie dust to create loosely coupled integrated applications. One of our aims with this new edition is to identify when loosely coupled works and when it doesn't, and to lay out the advantages and disadvantages of the different approaches to architecture. In consequence, our approach to IT architecture is very flexible and we present a range of architectural options such as middleware bus architectures, hub and spoke architectures, and Web services architectures. The common theme throughout is the notion of service-oriented architectures.

In concrete terms, describing Web services and other new technology has led to one new chapter and substantial changes to three others.

The third major area of change in this book has to do with design. As in the technology chapters, the changes in the design chapters are partly in response to changes in the industry. In particular, we have felt the need to discuss agile approaches to development and how they can cooperate with an architectural approach. But the wider reason for change is that the authors' understanding has developed in the intervening years. In particular, we have been able to describe a lightweight approach to integration design and to provide a better view on how the IT architecture supports business processes. These changes have led to one new chapter and substantial changes to two others.

We have also tidied up some of the material on changing existing systems and in the process eliminated one chapter. Also, two of the earlier chapters that discuss the history of middleware have been merged since some of the technology has sunk in importance.

Of course, with all these changes to the body of the book, we needed to rewrite the first and last chapters.

In spite of all the changes in the IT industry, it is gratifying to observe how much of the book has remained useful. Although there are substantial changes, we have retained the original structure. The first few chapters are about technology; the middle chapters are about using the technology to meet performance, resilience, security, and system management goals; and the last few chapters are about design.

## **Organization of this book**

You can read this book straight through or use it as a reference. This section explains the structure of the book. If you intend to use it for reference and don't intend to read it through first, we encourage you to read at least Chapters 1, 6, 11, and 16.

In addition to introductory and concluding information, the body of the book falls into four main sections:

- Middleware technology alternatives
- IT architecture guidelines and middleware
- Distributed systems technology principles
- Distributed systems implementation design

The thread that holds these topics together is the focus on IT architecture and implementation design.

## Introduction

**Chapter 1, The Problem**, is an introduction to the rest of the book. Using an example, it points out the main concerns of IT architecture.

## Middleware technology alternatives

**Chapter 2, The Emergence of Standard Middleware**, and the following two chapters are a historical survey of middleware technology. Chapter 2 covers remote procedures calls, remote database access, distributed transaction processing, and message-queuing technologies. It ends with a comparison of message queuing and distributed transaction processing.

**Chapter 3, Objects, Components, and the Web**, is an overview of the concept of object-oriented interfaces and components. The technologies discussed are DCOM, CORBA, COM+, and Enterprise JavaBeans. The impact of the Web is discussed and the chapter ends with an examination of the concept of sessions.

**Chapter 4, Web Services**, brings the history of middleware up-to-date with an overview of Web services technology, in the context of service and service-oriented architecture concepts.

## IT architecture guidelines and middleware

**Chapter 5, A Technical Summary of Middleware**, discusses middleware as a whole, beginning with the parts that make up middleware, with an emphasis on Web services. It also looks at vendor architectures, such as Microsoft .NET and Sun's J2EE, and at the issues surrounding middleware interoperability.

**Chapter 6, Using Middleware to Build Distributed Applications**, takes an application designer's view of middleware. It examines the major questions, what is middleware for and what is the role of middleware in architectural design? This chapter also proposes some basic architectural patterns—middleware bus, hub, and Web services architectures—and discusses the difference between tight and loose coupling of distributed applications.

## Distributed systems technology principles

These four chapters examine architecture design for nonfunctional requirements—performance, resiliency, and so on. We believe that the overall architecture of the system can greatly assist in providing good performance, resiliency, and so on. You cannot just rely on faster, more resilient hardware to get you out of any hole you happen to dig yourself into. These chapters explain why and give some solutions to common problems.

**Chapter 7, Resiliency**, explains the principles of resiliency in distributed systems. There is discussion on how to use backup systems effectively and on how to write applications so as not to lose vital transactions. **Chapter 8, Performance and Scalability**, explains the principles of performance and scalability in distributed systems. It describes the crucial importance of looking at the network traffic and the distribution of the data.

**Chapter 9, Systems Management**, deals with the principles of systems management in distributed systems. It discusses what centralized system management tools can and cannot do for you.

**Chapter 10, Security**, addresses the principles of security in so far as they impact on architectural design. The emphasis is on access control and where to implement it in the system. Web services security standards

are explained, especially the differences between Web services security and earlier forms of security.

## **Distributed systems implementation design**

**Chapter 11, Application Design and IT Architecture**, is about where architectural design belongs in the context of application design. It proposes an approach based on the idea of levels of design where each level includes some analysis to identify and correct flaws in the design. The following two chapters discuss the top levels in more detail. (Detailed design is well covered in many other books.)

**Chapter 12, Implementing Business Processes**, describes business processes and why they are important to IT architecture. Four large-scale designs for implementing business processes are discussed: Single Centralized server, Multiple Centralized servers, Pass Through, and Copy out/Copy in.

**Chapter 13, Integration Design**, discusses integration design at the task or dialogue level. A technique called task/message diagrams for highlighting integration design issues is presented, as are some integration design analysis techniques. The chapter ends with a further discussion of differences between loosely coupled and tightly coupled applications.

**Chapter 14, Information Access and Information Accuracy**, addresses two information topics that are particularly important for large-scale architectural design. (This book does not try to describe all aspects of design, and database design is well covered in many other books.) The information accuracy section is largely about whether to share data between applications or whether to implement some form of controlled data duplication. Retrofitting a data strategy on existing applications is also covered.

**Chapter 15, Changing and Integrating Existing Applications**, covers creating a new presentation layer for existing applications, integrating existing applications in a service-oriented architecture, and reducing reliance on batch processing.

## **Conclusion**

**Chapter 16, Building an IT Architecture**, summarizes the contents of the book and reviews a few case studies.

## **Boxes**

Throughout the book you will see references like this: (see the box entitled A box). They refer to the text set apart in screened boxes with a boldface heading. These boxes contain more information about the topic under discussion, perhaps a subject that is more technical than the body of the text or that is an esoteric area we couldn't resist writing about.

## **The Web site**

This book draws on many sources. One problem with the subject, as with most of IT, is the rapidly changing nature of the technology and the ideas behind it. We therefore decided that we would not put a list of sources and pointers to further information in the book itself, as there would no doubt be much additional material available within a few weeks of publication. Indeed, there would probably be new material at the time of publication!

We decided that the best way to provide readers with sources of information, and further case studies and examples, would be to use the Internet. Therefore, this book has an associated Web site, where readers can find additional material: <http://eCommunity.unisys.com/Books/ITArchitectures>.

On your first visit, you will be asked to complete a simple registration. You will then be forwarded to the appropriate Web page. This Web site will be maintained after publication to keep it up to date.

We would appreciate any comments you may have on the value of this Web site.

## Users Review

### From reader reviews:

#### Edward Gilbert:

The book IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) can give more knowledge and information about everything you want. So why must we leave a very important thing like a book IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition)? A few of you have a different opinion about book. But one aim that will book can give many data for us. It is absolutely appropriate. Right now, try to closer together with your book. Knowledge or information that you take for that, you are able to give for each other; you could share all of these. Book IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) has simple shape however, you know: it has great and massive function for you. You can look the enormous world by open up and read a publication. So it is very wonderful.

#### Stacey Thompson:

Reading can called imagination hangout, why? Because if you are reading a book especially book entitled IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) your mind will drift away trough every dimension, wandering in each and every aspect that maybe mysterious for but surely will end up your mind friends. Imaging every word written in a publication then become one form conclusion and explanation which maybe you never get just before. The IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) giving you one more experience more than blown away your head but also giving you useful information for your better life on this era. So now let us teach you the relaxing pattern is your body and mind will likely be pleased when you are finished examining it, like winning a. Do you want to try this extraordinary shelling out spare time activity?

#### Juan Crowe:

This IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) is brand-new way for you who has attention to look for some information because it relief your hunger info. Getting deeper you in it getting knowledge more you know or else you who still having little bit of digest in reading this IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) can be the light food in your case because the information inside this book is easy to get by means of anyone. These books acquire itself in the form and that is reachable by anyone, sure I mean in the e-book form. People who think that in e-book form make them feel drowsy even dizzy this book is the answer. So there is absolutely no in reading a guide especially this one. You can find actually looking for. It should be here for you actually. So , don't miss this! Just read this e-book style for your better life and also knowledge.

**Pablo Cowart:**

That e-book can make you to feel relax. This kind of book IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) was colourful and of course has pictures on there. As we know that book IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) has many kinds or variety. Start from kids until teenagers. For example Naruto or Investigation company Conan you can read and think that you are the character on there. Therefore , not at all of book are generally make you bored, any it makes you feel happy, fun and rest. Try to choose the best book to suit your needs and try to like reading in which.

**Download and Read Online IT Architectures and Middleware:  
Strategies for Building Large, Integrated Systems (2nd Edition) By  
Chris Britton, Peter Bye #6A7TVZ3JQD0**

# **Read IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) By Chris Britton, Peter Bye for online ebook**

IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) By Chris Britton, Peter Bye Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) By Chris Britton, Peter Bye books to read online.

## **Online IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) By Chris Britton, Peter Bye ebook PDF download**

**IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) By Chris Britton, Peter Bye Doc**

**IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) By Chris Britton, Peter Bye MobiPocket**

**IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) By Chris Britton, Peter Bye EPub**

**6A7TVZ3JQD0: IT Architectures and Middleware: Strategies for Building Large, Integrated Systems (2nd Edition) By Chris Britton, Peter Bye**