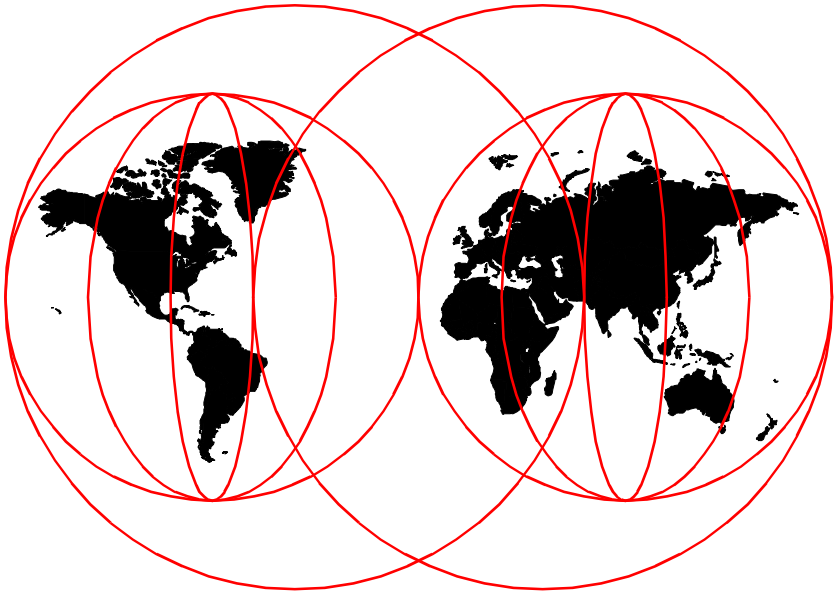
An Approach to Designing e-business Solutions

Don Bagwell, George Galambos, Randy Van Winkle, Adrian Walmsley



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5949-00

SG24-5949-00

International Technical Support Organization

An Approach to Designing e-business Solutions

December 1998

Take Note!

Before using this information be sure to read the general information in Appendix G, “Special Notices” on page 169.

First Edition (December 1998)

This edition does not apply to any specific operating system or application other than those described within this document.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

With the number of homes and businesses connected to the Internet increasing every year, many IBM customers will look to the Internet as a source of e-business opportunity, and look to you for help in designing e-business solutions and making purchase decisions.

This redbook will help IBM technical professionals who are faced with proposing or designing an e-business solution for a customer, but are not certain how to perform that task. It is intended to guide these professionals in mapping IBM's many e-business solutions to a customer's business and technology requirements. Some level of familiarity with e-business technologies is assumed.

Designing an e-business solution is not a precise process, so no step-by-step checklists are provided, nor does this redbook attempt to be a comprehensive source of technical and product information. However, it does present a broadly defined approach that, when coupled with information on how and why key IBM e-business technologies apply, helps you design and propose an e-business solution to your customer.

Note: This redbook was issued as an Internal Document in December 1998. No updates have been made to the original text.

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, Poughkeepsie Center.

Don Bagwell is a Senior I/T Specialist in the Advanced Technical Support organization in the Washington Systems Center in Gaithersburg, Maryland, USA. His focus is on supporting and promoting IBM's e-commerce solutions on the S/390 platform. He has 15 years of experience with IBM, starting as a logistics analyst at the Research Triangle Park, North Carolina manufacturing facility. A move to TPNS testing and development started his career in communications and technology. Five years in the field as a Systems Engineer and four years as a Sales Specialist selling IBM networking and RS/6000 solutions provided him with a customer's perspective on IBM.

George Galambos is a Distinguished Engineer, working as an IT Architect in the IBM Consulting Group in Montreal, Canada. He has 24 years of experience in the field of IT Architecture and Systems Design. He holds a

Ph.D. in Chemical Engineering from the Budapest Technical University. His areas of expertise include Infrastructure Design, Performance and Availability Analysis. He has contributed to several IBM Methods, including the Infrastructure Design Method and the Enterprise Solutions Structure. His current interests include system design for e-business and asset-based solution design.

Randy Van Winkle is a Certified IT Architect with the Global e-business Integration Team in Dallas, Texas, USA. He has 29 years of experience in the computer industry. With IBM, he has worked as a System Engineer architecting and designing solutions for customers, and as a Systems Engineering Manager leading a team supporting technical marketing. His areas of expertise include e-business solution architecture and design and integration to OS/390 systems, specifically with CICS applications. Based on his extensive experience architecting and designing e-business solutions, he has developed Web technology curriculum and provided instruction in Web technologies internationally.

Adrian Walmsley is a Technical Staff Member and Consultant IT Architect in the e-business Centre of Competence in the EMEA ISC in Hursley, England. He has over 30 years of experience in the IT industry. After a short spell teaching as a volunteer in Nigeria he took a degree in mathematics from Oxford University. His areas of expertise include the design and performance of large systems, Web technology in general and most recently e-business applications based on Domino. As an assignee to Poughkeepsie in the mid-1970s, he was involved in the inception of the ITSO redbook program, and since then he has contributed to a number of redbooks.

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

William G. White (Project Leader)
IBM International Technical Support Organization, Poughkeepsie

Sandra L. Ambrose
IBM Global e-business Integration, Dallas, Texas

Chris Deaves
Systems Architecture, IBM United Kingdom

Rod L. Fleming
IBM Global e-business Integration, Dallas, Texas

Romelia H. Flores
IBM Global e-business Integration, Dallas, Texas

Leonard S. Hand
IBM Global e-business Integration, Dallas, Texas

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 191 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

  For Internet users http://www.redbooks.ibm.com
  For IBM Intranet users http://w3.itso.ibm.com

- Send us a note at the following address:

  redbook@us.ibm.com

Chapter 1. e-business Solutions Approach Overview

Like any complex design project, some form of process is needed to go from the first idea to the completed design. That process might be completely subjective, as it might be when you prepare a meal, or it may be very complex and require formal project management. However, some process always takes place.

With an e-business solution design project, the process is more complex, because the range of options available is so great. This makes the task of describing how to design a solution that much more difficult. It would be unfair to you and your customer to present a simple checklist of activities, for each solution will be different from the last. Therefore, what this redbook does is provide an *approach* you can employ to design a solution. *There is no substitute for experience*, however, and this redbook assumes some experience in technology issues as are found in e-business scenarios.

In this chapter we describe the approach to designing e-business solutions that is provided in this redbook.

1.1 e-business Building Blocks

Throughout this redbook we use a diagram that represents a set of *building blocks* that are used to construct an e-business solution. This diagram is shown in Figure 1.

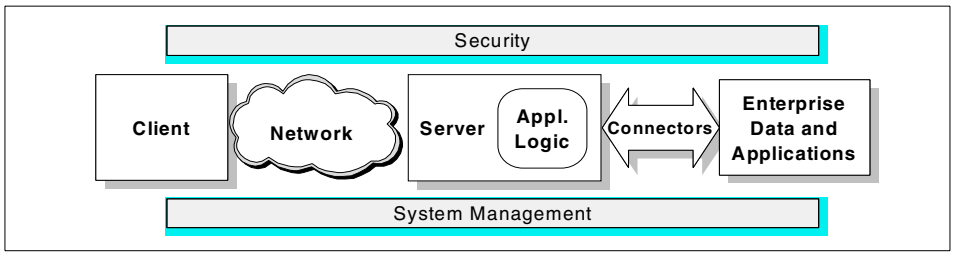


Figure 1. The Building Blocks of an e-business Solution

Admittedly, this is a very simple diagram. Real e-business solutions involve far more detail than is implied here. The value of using such a diagram remains, however. By using the building blocks in each chapter, we keep your focus on the basic issues of e-business solution design while we take you through the process described in this redbook.

Many such diagrams exist within IBM and the industry to illustrate the architectures and products used in e-business. One of these is the IBM's e-business Application Framework (ebAF), which is shown in Figure 2 on page 2.

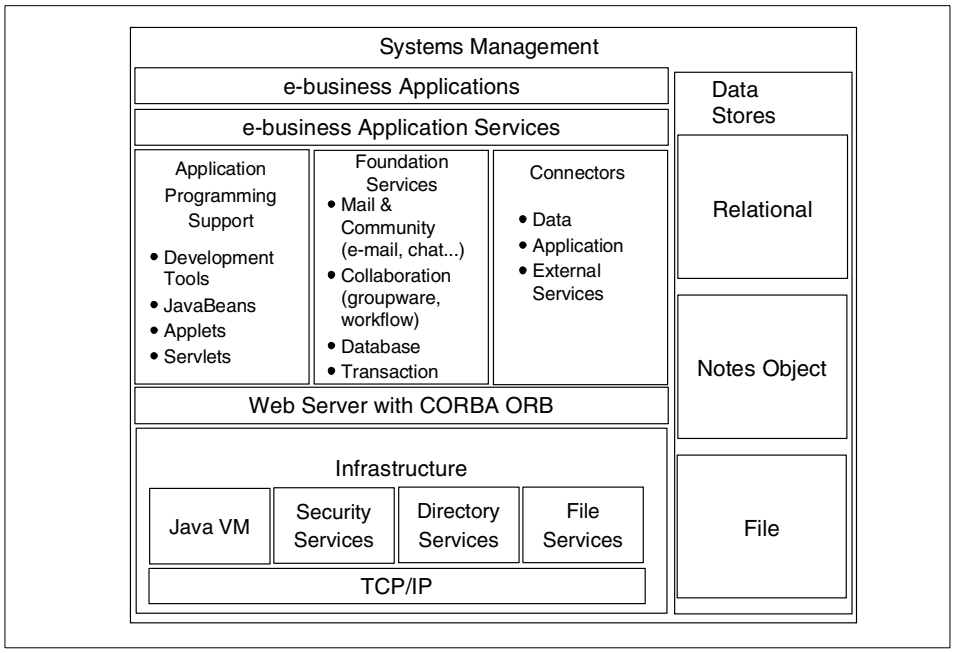


Figure 2. IBM's e-business Application Framework

IBM's major competitors in the e-business market, such as Microsoft, Netscape and Oracle, have similar diagrams. We cover the IBM e-business Application Framework and the competitor frameworks in Chapter 6, "Select Components" on page 95.

These more complex charts become more valuable as you gain knowledge in the technologies and products of e-business. However, as you start out, keep your mind focused on the simple building blocks.

1.2 Approach Provided in this Redbook

We are very deliberate in our use of the term *approach* rather than *methodology*. A methodology implies a formalized process with fairly well-defined entry and exit criteria at each step of the process. Within IBM, several such methodologies exist, such as WSDDM and SI/AD, complemented by the design assets documented in ESS (see Appendix F,

"Other Design Methods and Sources of Information" on page 163). They tend to be very thorough and complex; well-suited for what they are intended to do, but well beyond the scope of this document.

Instead, we borrow some of the key concepts from these existing methodologies to create a simpler, less formalized approach.

With this approach we discuss:

- Gathering the customer's requirements
- Developing a set of architectural alternatives
- Choosing one of the architectural alternatives
- Selecting components that address the alternative chosen

This approach is represented using a flowchart as shown in Figure 3.

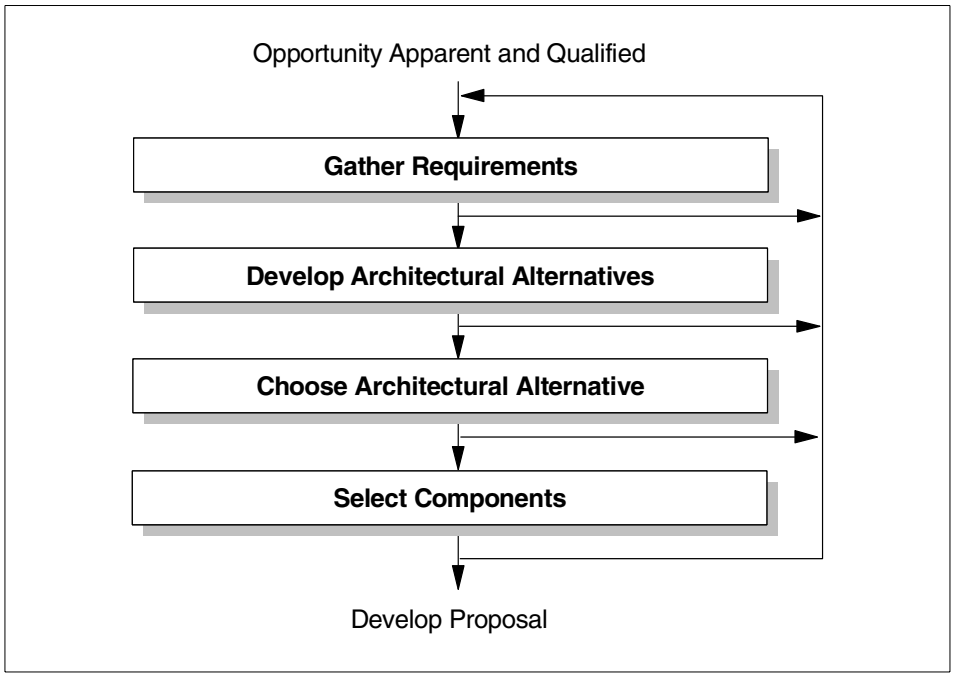


Figure 3. Flowchart of e-business Design Approach Activities

The arrows that loop back illustrate a very important point: *this is an iterative process*. If a significant new requirement surfaces after you have developed the architectural alternatives, you may need to go back and redevelop the alternatives. Similarly, if during the choosing of the architecture you find the

customer indicates none of the alternatives really meets their requirements, you will need to go back and re-gather the requirements.

Each of these phases is described in more detail in this chapter, and then in far greater detail in individual chapters dedicated to each step of the approach.

1.2.1 Gather Requirements

The requirements you gather act as the foundation upon which the rest of your solution will be constructed. It is these requirements that you map your architectural alternatives to, and it is these requirements that are the basis for the selection criteria used when choosing an architecture.

Requirements come in many forms: business, technical, personal and political, to name a few. In this phase of the approach, we focus on business requirements and technical requirements. The relationship you have with your customer will largely determine how well you understand the personal and political agendas that might be present.

Along with gathering requirements for the new e-business solution, you will also need to understand the customer's existing environment. New solutions are rarely developed completely separate from what is already in place.

Gathering requirements sounds like an easy thing to do, but it is not. It requires that you have knowledge of both the customer and the e-business solutions, and that you are skilled in asking questions and listening to the answers provided. It also requires that you use your customer's time efficiently.

What Is Provided

- An outline of a "workshop approach" to gathering requirements that has proven effective in real sales situations
- A set of sample requirements that indicate an e-business solution
- An "exit checklist" to be used to ensure you have your customer's agreement and that the requirements gathered are complete and accurate

This topic is covered in detail in Chapter 3, "Gather Requirements" on page 19.

1.2.2 Develop a Set of Architectural Alternatives

Some business problems can be solved using well-understood design patterns, such as those in ESS (see F.2.2, “Enterprise Solutions Structure (ESS)” on page 166). More often, several architectural alternatives will be technically possible. Presenting multiple alternatives is valuable because it provides a forum in which to discuss the technologies relative to the requirements. It is usually at this point that additional requirements or issues arise.

Being able to understand in detail all the architectural options available is a daunting task. It would be impossible to deliver that level of detail in a single document such as this, so instead we will focus on the approach and provide examples of some of the issues.

What Is Provided

- A discussion and example of a *business scenario* used to validate the requirements and the proposed alternatives
- A discussion of how to settle upon a few basic high-level architectures
- Information on how to select technologies for each building block, including a set of questions to help you understand the issues and reference information to help you select from the available technologies

This topic is covered in detail in Chapter 4, “Develop Architectural Alternatives” on page 39.

1.2.3 Choose One of the Alternatives

Once a set of architectural alternatives has been developed, the next phase is to decide which alternative is best suited to the customer environment. This is done by reviewing each of the requirements gathered in Chapter 3, “Gather Requirements” on page 19, and assessing, or *grading*, how well each alternative provides for the fulfillment of that requirement.

This phase, like others in this approach, requires some level of understanding of the architectures and technologies. What is important is coming to a consensus with the customer on which alternative will be the one upon which the implementation project will be based. By doing so, you focus the energy of the project on implementation, rather than competing between options.

What Is Provided

- An overview of a process by which the requirements gathered are mapped against the architectural alternatives
- A description of a proven approach to *grading* the alternatives on how well they suit each requirement, as well as an example of this grading process

This topic is covered in detail in Chapter 5, “Choose Architectural Alternative” on page 81.

1.2.4 Select Components

Prior to this point, we have been discussing *architectures* and *technologies*, not specific components or products. Once the architecture has been settled upon, the task of choosing the components begins.

There will be occasions where certain components will have been listed as a customer mandate, as discussed in 3.6.3.3, “Existing Conditions” on page 35. If, after an architecture has been settled upon, this mandate from the customer still exists, then it must be factored into the solution.

What Is Provided

- A listing of the common components available by building block
- Information you can use in your selection of the components
- Some assistance on where to go to gain further knowledge of specific components
- A set of guidelines by which you can determine if a Proof of Concept (POC) or Test for Indicators of Performance (TIP) is necessary

This topic is covered in detail in Chapter 6, “Select Components” on page 95.

1.3 Guidance

The approach represented in this redbook could consume a considerable amount of time and resource. That is why it is important to balance the value of the solution against the time and effort you will put into it. Many times the complexity of a solution is related to the business problem, not to its scale. To the extent that you can use proven designs as found in IBM’s store of Intellectual Capital (see Appendix F, “Other Design Methods and Sources of

Information” on page 163), you should. IBM has many practitioners skilled in these solution design methodologies and if the business problem appears very complex, it is recommended they become involved.

Chapter 2. What is IBM e-business

The purpose of this chapter is to provide you with a background on what is behind the term *e-business*. Many definitions exist within IBM. Generally they all revolve around the concept of information flowing across networks. With the Internet being such an accepted and pervasive vehicle for carrying information, it is often somewhere near the center of all definitions of e-business.

In this chapter we provide one such definition of e-business as found on IBM's internal Web site. That definition is comprised of three categories of solutions, each of which is described, with some examples, in this chapter.

Finally, an important aspect of e-business, the *business-to-business* segment, is described with examples to help illustrate how e-business encompasses more than you might think.

2.1 IBM's e-business Strategy

As IBM's "e-business Advisor" Web page (http://w3.ncs.ibm.com) puts it, IBM's unique competence for e-business is summed up with this simple phrase:

e-business = Web + I/T

What that means is that IBM's e-business strategy is aimed at taking advantage of the pervasive nature of the Internet to provide more people access to the valuable information contained in traditional Information Technology (I/T) applications.

Throughout this document, we use the diagram as shown in Figure 4 to represent this merging of the Web with traditional I/T. These are the building blocks of any e-business solution.

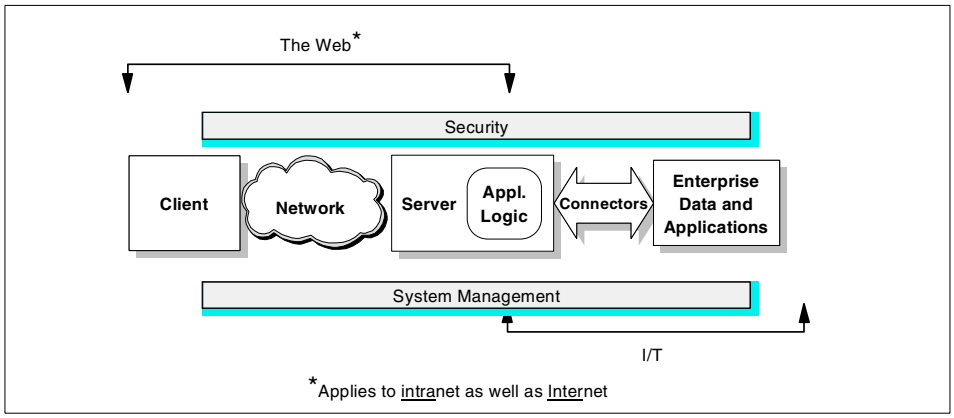


Figure 4. The Merging of the Web and I/T

This diagram is generic enough to apply to any e-business solution, though not all of the building blocks illustrated may be employed. For example, a simple static-HTML Web site would only consist of the Client, the Network and the Server. On the other hand, a solution that allowed customers with browsers to review their account status history maintained in a DB2 database would employ each of the blocks illustrated.

The diagram may become more complex when you incorporate a business-to-business scenario, where one company's servers talk to another company's servers across the Internet, as shown in Figure 5 on page 11.

Note that Security and Systems Management layers still apply, although they have been removed from this diagram to keep the diagram simple.

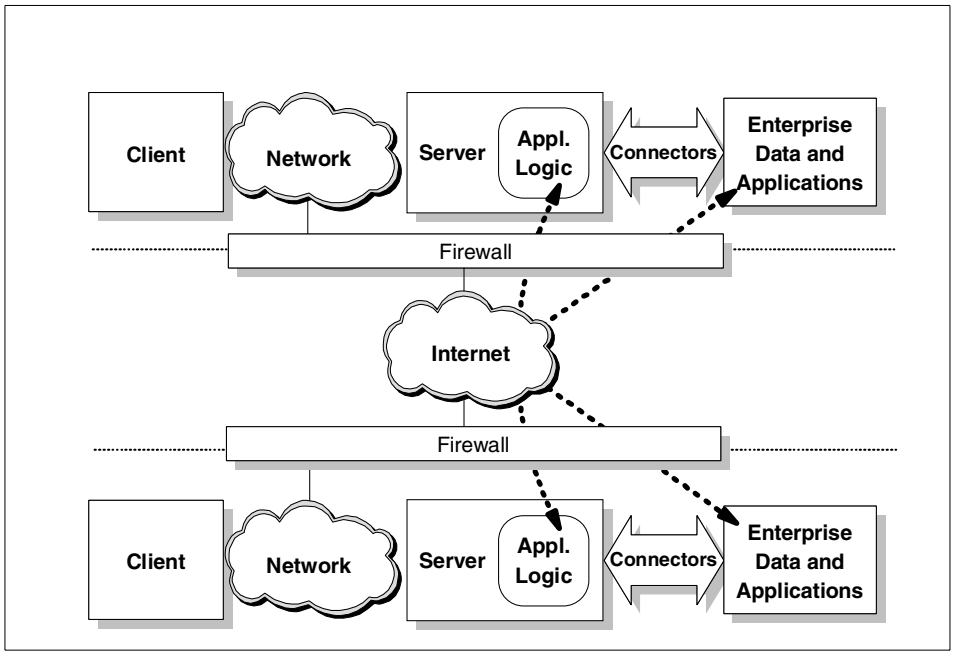


Figure 5. Application-to-Application over the Internet

The topic of business-to-business scenarios is covered in more detail in 2.3, “Business-to-Consumer and Business-to-Business” on page 13.

To make the IBM e-business strategy easier to understand, IBM has broken it down into three categories. Those categories are described in 2.2, “What IBM Means by the Term e-business” on page 11.

2.2 What IBM Means by the Term e-business

IBM has adopted the term e-business to refer to a broad range of products and services. The technology is changing very rapidly, and the terms used to describe it change quickly as well. To make this easier to understand, IBM subdivides e-business into three groups: *e-mail and Collaboration*, *e-commerce* and *Web Application Servers*. While there is some overlap of products between these groups, they are useful in loosely categorizing IBM’s offerings. Let us take a look at what each one means, and see examples of IBM products that fall into each group.

If you wish to explore these areas more fully, visit IBM's e-business home page on the IBM intranet at:

http://w3.ncs.ibm.com

2.2.1 e-mail and Collaboration

There is no shortage of information in the world today. What is not always easy is knowing where the information you desire resides, how to get to it, or how to share that information with other people.

IBM sees its e-mail and collaboration solutions as addressing this problem. The business benefit is greater productivity from people through easier access to information. This implies not just retrieving the information, but doing something with it. For example, a customer may wish to provide their employee benefit forms on-line. A simple Web site could do this.However, imagine the customer wishes the completed forms to be routed to the proper person in the Human Relations department based on the first letter of the employee's last name. Imagine further a system that tracks the completion of the form's processing, complete with an e-mail back to the employee that submitted the form. IBM's e-mail and collaboration solutions are aimed at doing exactly these kinds of things.

To most readers, the concept of e-mail is easy to understand. It was one of the very first uses of the Internet, and continues to be a very popular tool used by both individuals and businesses.

Collaboration helps people work together more closely, even though they may be miles apart. For example, the authors of this document used the IBM Lotus Notes "WorkRoom" feature to exchange ideas on the content of this redbook long before we actually arrived in Poughkeepsie, New York to start writing it. Even though we came from three different countries (USA, Canada and England), we *collaborated* using Lotus Notes.

2.2.2 e-commerce

The reasons for offering products and services for sale over the Internet are compelling. The opportunity to expand your customer base to customers you may never have considered reaching is one reason. The very real possibility of lowering your marketing and sales expense is another, and the list goes on.

To IBM, e-commerce is more than just a customer selecting a product from a on-line store page. To IBM, e-commerce includes all of these activities:

- Pre-sales activities, such as marketing and brand awareness.
- The sales transaction itself, which will likely include accessing information and processes on fulfillment systems, such as inventory control systems, pricing systems, and shipping/receiving systems.
- Processing of the payment, which may involve new technologies such as the Secure Electronic Transactions (SET) protocol.
- Post-sales activities, such as on-line warranty card registration.

Note that e-commerce includes not just business-to-consumer selling, but business-to-business transactions as well. That is a market projected to be many times larger than the market to sell goods and services to consumers.

2.2.3 Web Application Servers

Successful Web sites “do things”. They create a hub for customers and employees. Done well, they provide a new way to buy and sell, and to deliver information and new products. A good Web site can contribute to the company’s bottom line. This is the real promise of e-business.

The Web Application Servers solution group includes technologies that allow a Web server to do things. For example, a bank may wish to provide banking functions from its Web site. This would require something more than a simple static HTML Web page, as the information that is presented would need to be specific to the customer that has connected to the site. A connection with an enterprise server containing the customer’s account information would most certainly be required. The processing of all of this activity might take place on the same server that hosts the Web site, such as on a S/390 server.

Some of the technologies you should look for in an application server include support of enhanced server-side logic support, clustering, caching, profiling session management, and performance management.

2.3 Business-to-Consumer and Business-to-Business

In addition to the three solution categories previously described (see 2.2, “What IBM Means by the Term e-business” on page 11), the IBM e-business solutions can be applied to two very broad environments:

- Business-to-Consumer
- Business-to-Business

This is a useful discussion because many people’s first impression of e-business is that of consumers purchasing products from their home PCs, or people within a company accessing a collaboration server inside their

company. While that is a vital part of the story, it is not the whole story. For in the world of business, the flow of information and money *between businesses* often dwarfs that found between the business and the consumer.

In the past, a company's network was very much a closed and protected asset. Granting access to the network involved providing some form of dial connection, or in the case of large companies, some form of host-to-host connection. In any case, it was something that was done only after careful consideration.

The presence of the Internet provides a common, standardized connection between virtually any company in the world today, and one that opens up a world of opportunities. We provide three examples of such opportunities: one in e-mail and Collaboration, one in e-commerce, and one in Web Application Servers.

2.3.1 Business-to-Business e-Mail and Collaboration

The concept of e-mail between businesses is an easy thing to comprehend. After all, it has been going on for several years now, and a number of standards have been defined (POP3, IMAP4, MIME).

Collaboration is more sophisticated, and involves processing on the server to collect and manage the information that is being shared. Lotus Domino allows you to accomplish this within a company, as shown in Figure 6. Note that Security and Systems Management layers still apply, although they have been removed from this diagram to keep the diagram simple.

.

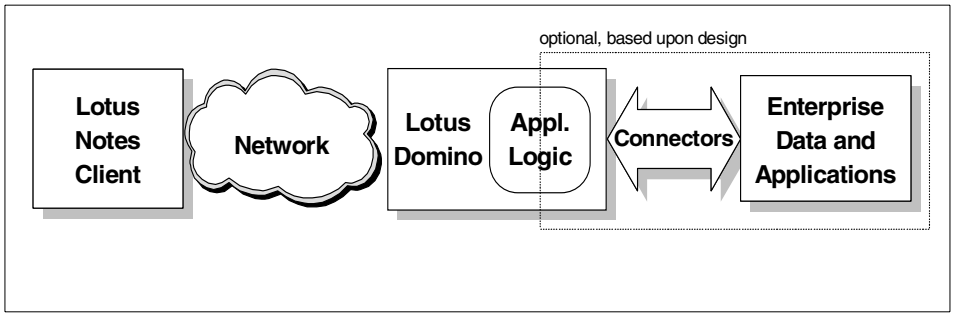


Figure 6. Example Collaboration Solution within a Business

However, this design can be extended so that users within another company, for example a supplier to your company, can access collaborative processes on your server as shown in Figure 7.

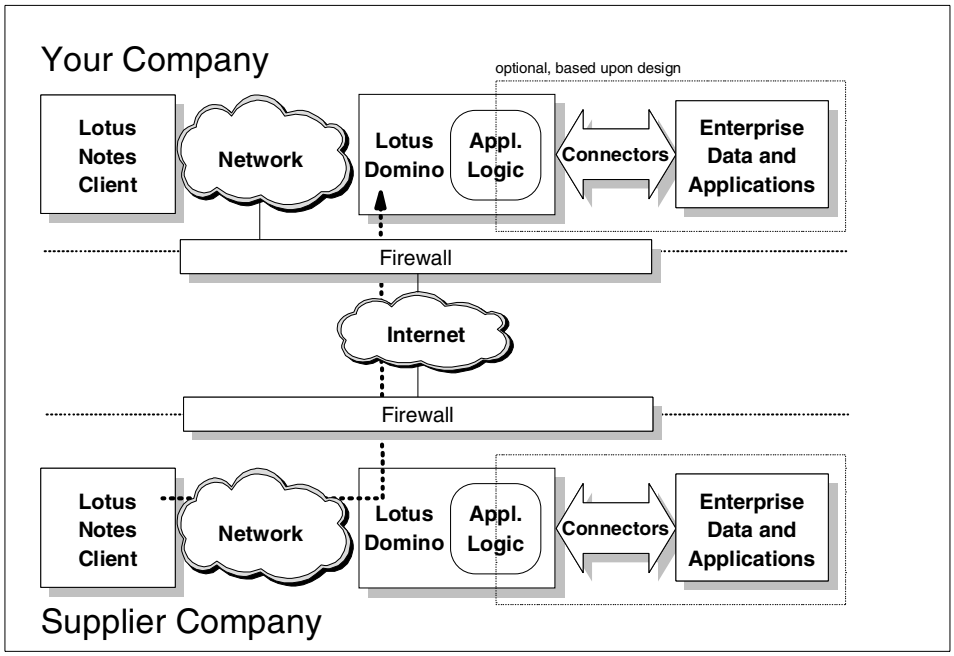


Figure 7. Collaboration across Company Boundaries

Imagine collaboration taking place about the design of a replacement component. In the past, this would have involved the sending back and forth of plans and revised plans. Maybe several on-site visits would be necessary to hammer out the details of the new part. Now, collaboration would allow much, if not all, of this activity to take place over the Internet.

This is not the place to go into great detail on exactly how this would be implemented. Let us simply leave it at this: company boundaries used to define the extent of collaboration, but that is not necessarily the case today.

2.3.2 Business-to-Business e-commerce

The amount of business that is transacted between companies in the United States alone is *huge*. Take the example of an automobile. The mining company sells ore to the steelmaker. The steelmaker sells the rolled sheet metal to automotive parts manufacturers, which construct the various parts. The parts manufacturers then sell the parts to the auto manufacturer, who

builds the car and then sells it to an independent dealer. The sum total of all of those transactions represents a great deal of money.

To some extent this business-to-business selling has been automated by Electronic Data Interchange (EDI). However, the potential exists for far more automation, particularly as the Internet has become so pervasive and connects virtually all companies, large and small.

Imagine a retail buyer wanting to purchase a supply of radial tires for a store in Columbus, Ohio. In the world of e-business, that buyer could access the tire manufacturer's server, check the supply on hand, and electronically purchase the supply desired. In this example, the picture might look like Figure 8.

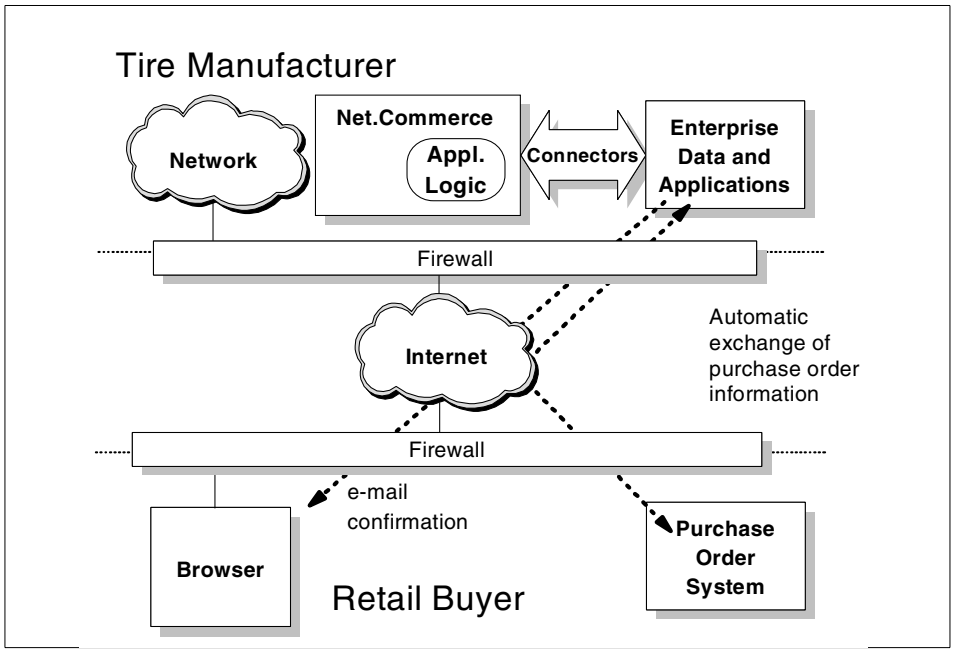


Figure 8. Business Buying from Business

The exchange of purchase order information could be accomplished across the Internet as well, or, if a system already exists, through a defined Electronic Data Interchange. The point is this: the buyer would not need to spend time exchanging phone calls with the tire manufacturer's sales representative to place the order or spend time generating a paper purchase order and mailing that to the manufacturer. What may have taken perhaps hours before now takes minutes.

2.3.3 Business-to-Business Web Server Applications

We described Web Server Applications as the group of technologies that allow Web servers to “do” things (see 2.2.3, “Web Application Servers” on page 13). In a business-to-business setting, this will most often imply some connection from the Web server to the enterprise data that already resides on production systems.

Take for example a freight company whose customers are other businesses shipping their goods around the country. A shipping manager at a company that uses this freight company might wish to check the status of a shipment as well as reserve ten rail cars for next week. All of this could be accomplished using technology we have described so far, see Figure 9.

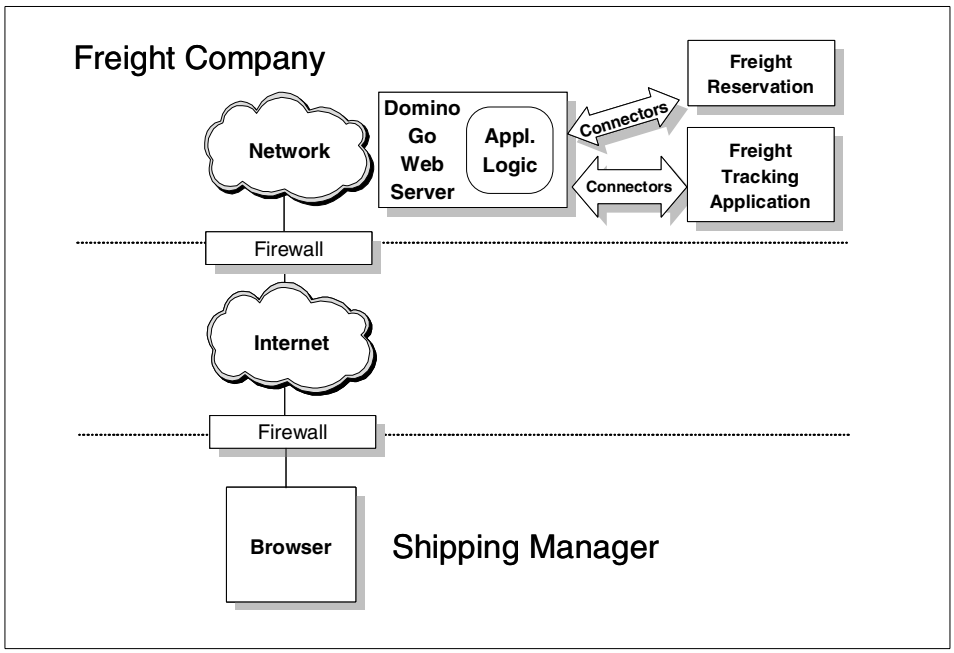


Figure 9. Using Other Company’s Web Site to Track Freight

As you can see, the application of e-business technology can be applied to almost any situation where people have a need to access information and do something with it.

Chapter 3. Gather Requirements

The importance of developing a good understanding of the business requirements cannot be overstated. Gathering requirements is not a process that occurs only once, but rather is an iterative process that may recur several times as the e-business solution evolves through the phases described in this document. Remember, the business requirements drive the e-business solution. Do not allow technology to drive the solution.

There are multiple techniques that can be used to gather the customer's business requirements. If a customer has an opportunity identified, then some of the requirements may already be known. The challenge is to bring these requirements forward, analyze them and prioritize them so that the process of identifying architectural alternatives can begin.

Associated with gathering business requirements is gathering a set of other requirements. These other requirements could include obstacles, existing conditions and functional or technical requirements that the customer has already decided on and which must be adhered to in the solution (for example, perhaps Oracle is the relational database that must be used, or Windows 95 is the client platform). You may need to deal with these because your customer may have already decided on them as a company standard.

Presented here is a list of questions that can be reviewed to insure all areas of the opportunity have been explored in terms of business requirements, as well as a description of a workshop approach that can be employed to "discover" requirements, prioritize them, and gain customer "buy-in". We also discuss understanding the current environment, as this can create some of the biggest obstacles in developing an e-business solution.

3.1 Overview of Requirements Gathering Process

Figure 10 illustrates the high-level approach represented in this chapter.

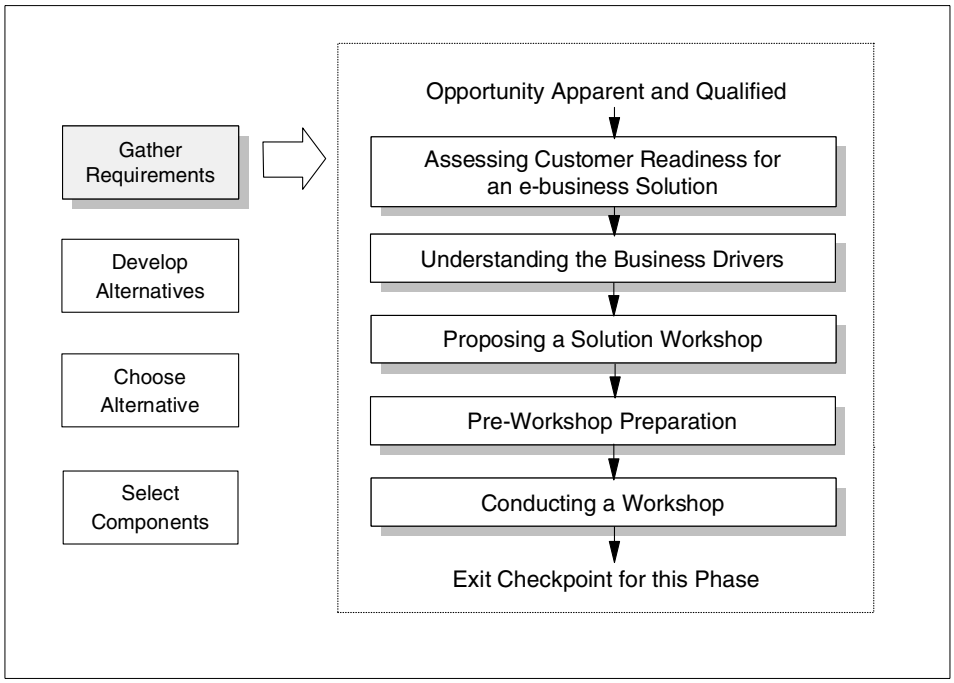


Figure 10. Overview of the Requirements-Gathering Phase

One final point about this flowchart: it is assumed that some form of e-business opportunity is apparent and you have qualified the customer's interest in exploring it further. Proposing a Solution Workshop before a customer has committed to the notion that a solution is needed will usually result in failure. Therefore, do not use this technique when you are in the interest-generating phase of a sales cycle.

3.1.1 Preliminary Requirements That Indicate an e-business Solution

Some requirements voiced by customers early in the sales cycle indicate that an e-business solution may be what they are seeking. When you spot these requirements, continue to ask questions to get a better understanding of what your customer is looking for. A brief explanation of the e-business opportunity behind the requirement is provided to help you qualify the opportunity and proceed to the gathering of requirements that leads to a solution design.

3.1.1.1 e-mail and Collaboration Requirements

- *Share information between groups or departments:* If your customer finds that different groups within the company tend to duplicate their efforts because they are not aware of what the other is doing, then a collaboration solution will help overcome this without imposing the drudgery of meetings and status reports.
- *Reduce the cost of planning for projects:* Projects often require planning in the early phases that involves travel, which is costly. An effective collaboration solution can reduce this cost while still maintaining the flow of information between participants.
- *Overcome the obstacles of teams in different parts of the world*: Organizing status meetings, even by phone, is difficult when team members are located across the globe. Collaboration solutions can reduce much of this effort by allowing participation on each members' schedule.
- *Reduce the processing of paper forms and processes*: Processes based on paper involve delay and cost. Collaboration solutions can replace paper-based forms with electronic forms, and route and file these forms based on any number of conditions.
- *Have better control over who accesses information*: Simple Web sites can offer information to users, but determining exactly who is accessing the site is difficult. User ID and password schemes can be employed, but do not offer complete protection against fraud. Lotus Notes, as an example of a collaboration solution, uses digital certificates that offer a more robust form of security. The server can be more certain of the true identify of the person accessing the site with such technologies in place.

3.1.1.2 e-commerce Requirements

- *Expand the base of customers*: Because a business presence on the Web is available to people from all over the world, more people will have an opportunity to see and consider doing business with that company.
- *Have hours of operation that are 24 x 7:* Because an e-commerce solution can be an automated process, you now have the opportunity to stay open for business 24 hours a day. If your customer base is now the entire world, staying open 24 hours a day becomes critical.

  Note: A 24 x 7 solution means more than simply keeping a Web server up 24 x 7. If connections are made to enterprise applications or data, those systems must also support a 24 x 7 operation window. All of this implies that a careful study of what 24 x 7 means in the customer's setting should be conducted before a 24 x 7 solution can be assured.

- *Reduce the cost of a sales transaction*: To the extent that you can automate the sales process through an e-commerce solution, you can

start to reduce the cost of that sales transaction. This is particularly true of mundane sales transactions where the customer knows what they want.

- *Provide better pre-sale and post-sale support to customers*: e-commerce solutions can provide presale information that is important to customers, and can help customers enjoy and use their purchases after the sale.
- *Make the existing Web site more dynamic in nature:* Many early Web implementations consisted of static HTML pages. This becomes very difficult to manage if the number of pages gets too large. An effective e-business should be largely dynamic, taking advantage of technology that automates this process rather than relying on manual processes.
- *Tie the existing Web site into existing enterprise systems:* Any existing Web site that relies on the manual duplication of data from another system is one that can be improved. Most of the business data in the world today exists in enterprise servers that can be connected to the Web servers to make this process far more effective.
- *Employ new methods of payment over the Internet*: Newer payment technologies such as Secure Electronic Transactions (SET) can provide greater security and integrity to the Web site wishing to use them. A requirement to use these new technologies suggests a customer who has concerns about their own business security, or believes more business can be generated by employing the technologies.

3.1.1.3 Web Application Server Requirements

- *Tie a Web server into existing enterprise systems*: A Web site that stands alone is one that will not do much good. If a customer wants a Web site to perform real function, as opposed to only offering static documents, some form of connection to other existing systems will be required. Since most of the business data in the world exists on enterprise servers, connections to those systems are imperative.
- *Process requests on the Web server, not just serve static pages*: Often a user will request that something be done based on the clicks of their mouse. This processing can be done entirely on the same server as the Web server, or it can be done on a different processor. How you architect this is in large measure what this redbook is all about.
- *Provide customized reporting to users from a Web site*: Offering content based on a user's identify, or a user's request, provides value from a Web site. This requires processing logic at the Web site. Web Application Server solutions can be used to provide this function.
- *Update the content of a Web site automatically, not manually:* Countless hours are spent every day by people manually updating Web sites. This update process can and should be accomplished automatically wherever possible.

- *Provide good performance and ability to scale the server:* The Web Application Server should provide good performance and ability to manage performance with techniques such as support for caching, clustering and load balancing.
- *Providing session management capability:* Web application developers should not spend valuable time worrying about how to maintain sessions within the application. The Web Application Server should provide these services.

3.2 Assessing Customer Readiness for an e-business Solution

You should determine the readiness of the customer to accept an e-business solution. This is not a formal checklist activity. It is more an informal assessment based on your experience with the customer.

Knowledge of the customer's readiness will help shape the education you provide in the Solution Workshop (see 3.4, "Proposing a Solution Workshop" on page 25), as well as shape the final solution design. A customer who is still running SNA over a coax connection might be a candidate for a slower, phased approach, while one with more advanced Internet skills might be ready to accept a more aggressive solution.

This assessment, therefore, would include the following:

- The level of technical skills, including:
  - Web server
  - Object technologies
  - Java
  - Application development tools
  - Database skills related to distributed data
  - Transactional skills
  - Maintaining state in a Web application
  - Security
  - Standards
- The attitude of the decision maker towards new technology adoption.
- The degree of risk the customer is willing to support.

The attitude and degree of risk will have an effect on the architecture of the solution. Some customers are reluctant to move quickly on new technology adoption, even if the technology is proven in the marketplace. Such a reluctance can result in the adoption of an ambitious design. Similarly, an ambitious design might entail some degree of risk of failing, or of costing more than originally planned. A customer unwilling to assume much risk will

tend to shy away from such a design. You must factor into your design decisions your assessment of the customer's willingness to adopt new technology, and the customer's willingness to accept risk.

This assessment should be in the form of a simple rating of high, medium or low.

Armed with this information, you can determine the level of detail needed in discussing the technologies. Reassessment of these readiness factors should be done as the design process continues.

3.3 Understanding the Business Drivers

Technology is seldom desired for its own sake. Rather, some need or problem or opportunity presents itself, and those responsible for the business begin to seek out a solution.

The forces that drive the business are very important to understand when designing a solution. They form the basis from which other requirements derive, and provide the justification for the eventual solution.

The objective of this chapter is to gather sufficient requirements to develop several architectural alternatives for consideration. Business requirements themselves do not usually equate directly to technological solutions, but questions about the business drivers offer a valuable tool to uncover requirements that can be used in the solution design.

Remember that we started with the assumption that an e-business opportunity had been identified and qualified. Though a design is not yet available, some early notion of a solution is typically in the mind of the customer, such as "Make our customer order information available from the Internet." To further refine the customer requirements ask your customer this: "How will this e-business solution...

- increase the number of customers?"
- lower operating costs?"
- enhance the image of the company?"
- get your products to market faster?"
- make your business more competitive?"
- reduce your development cycle?"
- increase revenue or decrease costs?"
- enhance customer relationships?"
- improve inventory and procurement management?"
- improve channel relationships?"

- improve customer service?”
- make your employees more effective in teaming and collaboration?”
- allow you to reach new markets?”
- reduce distribution costs?”

This is a conversation best had with an executive of your customer, rather than someone from the technical I/T staff. Often, the answers will be phrased in terms of things apparently unrelated to e-business technology.

From this, you will derive a list of things the executive has on his or her mind, such as the cost per sale to their customers is above the industry average, or that their customers have complained that post-sales support is too difficult. These are the *business drivers*. These are what your customer is looking to address with an e-business solution.

When you have developed this list of business drivers, review them with the executive and gain agreement that they are complete and accurate. Later, in the "Solution Workshop", you will review this list with both the business and technical representatives present to make sure the business drivers are understood.

Note that being aware of what other IBM customers are doing in the same industry can help you better understand your customer’s business problem. IBM’s e-business Advisor provides technical references by industries (see: http://w3.ncs.ibm.com/pcid.nsf/Industry?OpenView).

3.4 Proposing a Solution Workshop

A Solution Workshop is simply a gathering of the key business and technical people, from both the customer and IBM, who will have a say in the final solution design. The goals of a Solution Workshop are:

- To understand the existing environment and requirements
- To review the business application
- To provide education where appropriate
- To clarify the objectives, both short and long term
- To identify the issues
- To discuss technical alternatives (pros and cons)
- To recommend the next logical steps

A workshop may run anywhere from half a day to several days, depending on the complexity of the environment. Do not expect this to be a two-hour affair.

3.4.1 Why a Workshop

Getting everyone in a room together for a workshop affords many advantages over attempting to work through this requirements-gathering process with people individually:

- Time is saved by having everyone present.
- With everyone present, the technical people can hear and understand the business people's perspective, and the business people can hear and understand the technical people's concerns.
- Often competing views from within the technical community can be aired and resolved, which leads to consensus on the final solution.
- Experience has shown this is the most effective way in which to accomplish the task of getting to a solution design.

The challenge of a Solutions Workshop is that it represents an investment in the customer's time, particularly when a number of their key people need to be brought together at the same time.

Therefore, your first order of business is to propose and secure permission from the executive sponsor of the e-business opportunity to hold a Solutions Workshop.

3.5 Pre-Workshop Preparation

Before a Solution Workshop can be performed, some up-front work needs to be done:

- Identify participants from the customer.
- Identify participants from IBM.
- Communicate the goals of the workshop to everyone.
- Gather documentation from the customer on their current environment.
- Develop the Solution Workshop's agenda and distribute to all of the participants.

In the following sections, each of these items is discussed in greater detail.

3.5.1 Identify Participants From the Customer and IBM

Who participates in the Solutions Workshop is the most important aspect of a successful workshop. A cross-functional team from both the customer and IBM is the key. As illustrated in Figure 11, the customer brings to the table information and knowledge concerning its business, applications and current

environment, while the IBM team brings knowledge of e-business technologies and experience in developing successful e-business solutions.

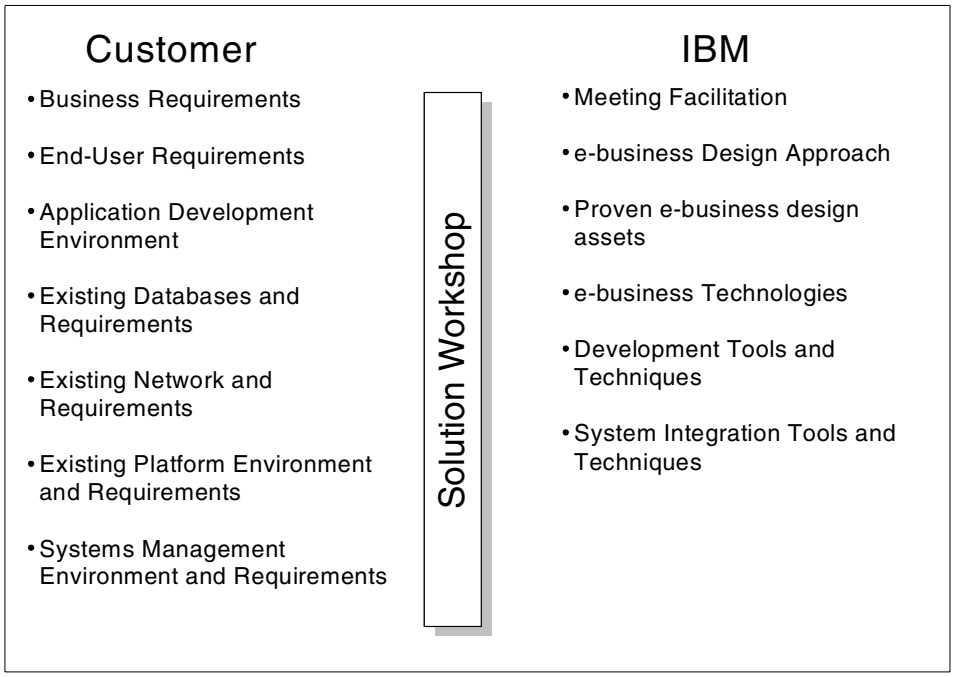


Figure 11. Skills Needed in a Solution Workshop

The total number of people in a workshop should be kept to a manageable number. Our experience has been that the customer should supply between four and six people, and IBM should supply around four people. No more than a dozen people should be at the workshop.

The person from the customer who represents the business requirements should be capable of making decisions regarding the final requirements.

Do not underestimate the value of a good meeting facilitator. That skill is listed at the top of the IBM list for good reason: a skilled facilitator will keep a meeting on track, and help the group reach consensus.

3.5.2 Communicate the Goals of the Workshop

As mentioned in 3.4, “Proposing a Solution Workshop” on page 25, the goals of a Solution Workshop are:

- To understand the existing environment and requirements

- To review the business application
- To provide education where appropriate
- To clarify both short-term and long-term objectives
- To identify the issues
- To discuss technical alternatives (pros and cons)
- To recommend the next logical steps

Capture these in a document, identify the executive sponsor of the workshop, and distribute these goals to the identified participants.

3.5.3 Gather Documentation on Customer's Environment

Information about the customer's current environment that will be pertinent to the e-business system or architecture design plans should be identified and documented *before* the workshop is conducted. This allows both the customer and IBM adequate time to understand and prepare for the workshop. The information should include known business requirements, information on the current environment, and any known restrictions.

Table 1, 'Documentation on Customer's Environment' describes the type of documentation that can be helpful:

Table 1. Documentation on Customer's Environment

Documentation	Comments
Target application overview (this may be high-level or sketchy at this point in the process).	Provides a way to communicate with IBM experts and a reference to look for existing design assets.
Existing network logical and physical design.	Assists in understanding what e-business technologies are best used on the existing network, which may have capacity restrictions or other factors that need to be taken into account.
Existing databases and database design.	Helps with development of connection between Web-based servers and back-end systems, which typically involve database access.
Existing Web environment (servers, firewalls and so forth).	Indicates the Web maturity of the customer and gauges the scope of the solution.

Documentation	Comments
Existing server environment including configuration diagrams, software versions, and future software upgrades.	Helps with discussions of leveraging existing infrastructure versus new.
Security requirements.	New security facilities are key to many e-business solutions, which cannot be developed without existing security policies.
Operational parameters (24 x 7 and so forth).	The expectation of 24 x 7 service over the Web puts focus on the operating schedule of the existing systems.
End-to-end performance requirements.	If an end-to-end service level agreement is in place, it is best to understand that before a design is proposed. End-to-end performance across the Internet cannot be guaranteed. Certain e-business technologies can also consume too much of this end-to-end time and therefore not be desirable.
Existing standards (network, naming, protocols and so forth).	Provides an indication of what some of the fixed parameters may be in the solution design.
Organization charts with skills.	Places customer participants in workshop within the context of their organizational structure.

3.5.4 Understand the Current Environment

Understanding as much as possible about the current customer environment is key to having a successful workshop. The IBM team should review the materials received from the customer. The workshop provides the forum where any questions or clarifications can be made.

Experience has shown that with good documentation from the customer, understanding the current environment can be rather straightforward. However, areas where problems can arise are usually associated with the existing network infrastructure and integrating with existing enterprise application and data.

3.5.4.1 Understanding the Network Infrastructure

An assessment of the current network infrastructure should be conducted. Table 2, 'The Customer's Current Environment' shows what information is necessary:

Table 2. The Customer's Current Environment

Information	Comment
Network baseline information that includes bandwidth, current network traffic, response times and so forth	Low-speed remote links could impede intranet solutions. Internet application could add substantial load to the corporate network.
IP design and facilities (DNS, VPN), if IP network exists	Existing IP network experience will expedite the implementation of e-business solution.
Network topology diagrams, including router and switch placement, sources and sync points	Network isolation, address space separation are important elements of e-business solutions, and thus impact their complexity.
Application/Data/Object placement principles and actuals	Mature organizations tend to have placement principles, which may be mandatory for the proposed solution. Also, many e-business opportunities demand access to existing applications and data.
Management and security policies	These may enable or restrict the use of certain technologies or components. For example, a firewall may restrict traffic to port 80 only, which may prohibit the use of applets that use different ports for communication.
The role of the Internet in the solution	The Internet brings with it a whole list of security and performance issues. Those issues need to be incorporated into the solution design if the Internet is utilized.

Depending on the level and quality of the documentation from the customer, this activity could be included in the Solution Workshop or may require a separate workshop. This may uncover the need for additional services such as developing a baseline, troubleshooting, network design and so forth.

3.5.4.2 Understanding Integration Requirements

Providing connections to the existing servers to get access to existing business functions and data is relatively easy. There are a number of IBM and

non-IBM middleware products that provide connectivity. The difficult issue is understanding the existing business functions and data structures so that the correct decisions can be made about which type of connectivity technology should be employed. If the customer is looking to integrate a new application with existing applications, the following issues should be explored:

Table 3. Customer's Integration Requirements

Issue	Comment
Need for real-time access	This will influence the selection of middleware used for access to the existing application or server. Real time *generally* implies a synchronous, logical connection to the server. Non-real time would permit the use of asynchronous connections used with message queueing technologies.
Ability to modify existing applications	The existing applications may be off-limits to modification. This may be the case because: the source code is no longer available; is a vendor product and therefore never open for modification; the application is stabilized.
Design of the existing application	The existing application's design will indicate how the connection can be made. For example, an older 3270 CICS application might not be open for modifications, so some form of interface to the 3270 layer of the application would need to be used to access the application. However, if the application source code is open for modification, then a program-to-program connection might be possible with EXCI or ECI. Also look at how the presentation layer is abstracted from the logic and data layers. Older applications typically employed programming techniques that blended the three layers together. To the extent the layers are logically separate in the application's design, the more flexibility you probably have in making the connection.

Issue	Comment
Existing infrastructure	Evaluate the type of middleware that presently exists (if any), and determine the customer's attitude towards maintaining that middleware or moving to a new technology. For example, in the late 1980s *screen scraping* was a popular technique by which data from an existing application was accessed and passed to a graphical interface. A customer might have a very large investment in that type of infrastructure, and might wish to maintain all or some parts of it.
Programming skills	Connecting a new application to an existing one will almost always involves some level of code customization, in the new application, or in the existing one, or both. A customer's programming skills are a valuable resource and your solution design may need to use that existing resource rather than stipulating a new programming skill. That being said, many technologies, such as Java, require object programming skills. If a customer has no such skills in-house, you might wish to phase a design in over time, or make the case to utilize contract skills to complete the project.
Data usage	The amount and type of data that needs to be sent and received from the client is an important factor. If the customer's network infrastructure is robust, the amount becomes less of a factor, unless some of the clients are remote and across dial lines. Similarly, the form of the data might need to be modified as it travels from data source to presentation point (the client, in other words). If the data is on an S/390 host, EBCDIC-to-ASCII conversion will need to take place, which can be challenging for binary data types such as packed decimal, floating point and signed fields. Also, the issue of national language support (NLS) comes into play if the clients will be located in different countries.

Because integration requirements can be complex, this activity might require someone on your team to work with the customer for several days. This activity would investigate these areas to come up with an assessment as to

the best way to interface with the existing systems. This could, potentially, be a customer-billable activity.

3.5.4.3 Understand Target Client Environment

The client environment will have an impact on the technology options you can propose. At issue is the degree of control the customer has over the clients. If the target environment includes clients not under the customer's control, the customer may not be able to specify things such as:

- The configuration of the client
- Issuing code updates to the client
- Testing of new releases of the client code
- The use of applets (Java, ActiveX) or other code

For example, the use of Javascript might be restricted because of the way different browsers behave with Javascript. Similarly, regarding the distribution of code updates or the use of Java applets, firewalls placed between your customer's server and the clients may prevent the transport of anything more than HTTP.

3.5.5 Develop and Distribute Workshop Agenda

You should distribute to all the participants the planned agenda for the workshop *prior* to it being conducted. At a minimum, you should plan on doing the following in a workshop:

- Review the goals of the workshop (this sets the stage).
- Review the business drivers behind the desire for an e-business solution (this level-sets everyone and positions the business needs as the true drivers of the solution).
- Review the current environment (this allows everyone to agree on the base assumptions as the solution design goes forward).
- Provide some level of education on the technology (not products) of an e-business solution. Not everyone will be familiar with this technology, so now is a good time to provide this.

  Note: It is too early in the cycle to know what products might be appropriate, so do not give in to the temptation to sell at this point. Be patient.
- Gather more detailed requirements of an e-business solution, see 3.6.3, "Solicit Requirements" on page 34.

3.6 Conducting a Workshop

The workshop itself is used to go through the entire approach described in this redbook, all the way from gathering requirements to the selection of an architecture. However, the aspect of the Solution Workshop covered in this section focuses on only the first block in our original flowchart (see Figure 3 on page 3).

3.6.1 Review Initial Understanding

As a starting point to the workshop, you should review the following items with all in attendance:

- The goals of the workshop
- The business needs and drivers (ideally, you would have the person responsible for these present this review)
- The current environment as IBM understands it from the customer documentation collected

The simple review of these items may very well generate some useful discussion which may help to further refine requirements.

3.6.2 Checkpoint on Items Reviewed

Before you proceed, make sure that everyone in the room is in agreement on the three items reviewed in 3.6.1, “Review Initial Understanding” on page 34. With everyone’s agreement, the foundation is now set.

3.6.3 Solicit Requirements

Because the workshop covers a broad range of topics concerning the customer's application, the opportunity exists for a single workshop to extract *other* requirements associated with the implementation of an e-business solution.

As you collect these other requirements, do not challenge them as they are offered; simply collect them. You will prioritize them later, as described in 3.6.5, “Prioritize Requirements” on page 36.

You may find disagreement between people from the customer over what constitutes a requirement. These discussions are valuable sources of information about your customer’s environment. Listen carefully, but do not allow the discussion on any given point to drag on too long. This is where having a good meeting facilitator becomes so important.

These other requirements can be broken down into the following categories:

1. Functional requirements
2. Obstacles
3. Existing conditions

3.6.3.1 Functional Requirements

These are the functions needed in the application. For example, if the application has the requirement to allow the user to navigate through multiple views of data related to claims information, the complexity of this requirement may dictate that the architecture accommodate a complex set of *widgets* (small icons on the screen) that control presentation. This may lead to a Java applet solution requirement versus an HTML presentation.

3.6.3.2 Obstacles

During the discussions, it may become apparent that some of the participants may have already decided that another vendor's approach should be employed. For example, there may already be a movement to use Microsoft's COM/DCOM technology to support implementation of an object-oriented application. Uncovering this early in the process can provide valuable information and help develop a marketing plan which could persuade the customer to use standard technologies.

Again, capture these requirements and ask questions to understand the *why* behind the requirement. Do not challenge it at this time. The requirements will be prioritized in 3.6.5, “Prioritize Requirements” on page 36, and the requirement may fall away.

Later, when you develop the architectural alternatives (see Chapter 4, “Develop Architectural Alternatives” on page 39), one of the alternatives might support a COM/DCOM strategy. In this way, you can compare and contrast the advantages of one alternative versus another so that during architecture selection it becomes easier to eliminate the competitive alternative with complete customer support.

3.6.3.3 Existing Conditions

Normally, a customer will have a set of existing conditions that have already been decided upon. The workshop gives an excellent environment in which to test these existing conditions.

For example, if an existing condition is the use of NT as an application server, a test of the commitment to this should be made during the prioritization of the requirements. If it is a soft commitment, then perhaps the ultimate

solution could be deployed on an IBM platform. If the existing condition is firm that NT will be the server, then this information will help shape the architectural alternatives.

Some examples of existing conditions are:

- That a specific set of application development tools be used
- That Windows NT be the application server platform
- That TCP/IP must be the networking protocol
- That certain industry standards must be followed
- That Sybase is the distributed database of choice
- That the client workstation will be Windows 95 or 98
- That the solution must be an object-oriented design

3.6.4 Checkpoint for Requirements Gathered

This process is clearly iterative. Before you move on to the prioritization of the requirements, you should ask whether any more requirements exist.

3.6.5 Prioritize Requirements

The process of gathering requirements will yield many items. The only way to meaningfully deal with them is to prioritize them. Experience has shown that prioritization should be kept simple and accomplished quickly. The recommended approach is as follows:

1. Get agreement from everyone to prioritize the list of requirements.
2. Go quickly through each item in the list of requirements, restating the requirement and making sure everyone agrees that it is still a requirement. Many times this will cause a requirement to be discarded or consolidated.
3. Go back to the top of the list and ask everyone for their agreement that the requirement has a priority of either high, medium or low.
4. If you end up with many high priority items, you may wish to sequence the ones marked high into an order of importance.
5. Gain agreement from everyone that this prioritized list is accurate and represents the needs of the functional areas of the business.

   Note: If you do not get agreement, then you must go back and keep working on the list until you do gain agreement.

The requirements marked high (and perhaps some of the mediums) become the things that drive the architecture and the final e-business solution.

Table 4, 'Example of Ranked Requirements' shows an example of what a set of ranked requirements might look like:

Table 4. Example of Ranked Requirements

Requirement	Rank
Business Requirements	
Reduce cost of project planning	High
Reuse information and knowledge between groups	High
Utilize market leader's technology	Low
Adhere to industry standards	Med
Other Requirements	
Use TCP/IP on network	High
Desktop system used for multiple applications	High
Use Windows NT as server platform	Low
User existing Ethernet network	Med
Use C++ as programming language	Low
Response time end-to-end must be less than 2 seconds	Med
User access must be secured	High
Access to data must be by user identification	High
Access must be dialup as well as LAN	Med

3.7 Chapter Checkpoint

At this point you are well into your Solutions Workshop, and you should have a prioritized list of requirements. Before you move on to developing architectural alternatives, ask yourself these questions:

- Do you believe that all of the business needs and requirements are properly understood and agreed to by all of the participants in the workshop?
- Do you believe that you and the IBM team fully understand the customer's current environment?
- Do you believe the requirements you have gathered so far in the workshop are complete, accurate, understood, properly prioritized, and agreed to by all the participants in the workshop?

- Is everyone in the workshop in agreement that you may move to the next step?

If you can answer yes to each question, then move on to Chapter 4, “Develop Architectural Alternatives” on page 39. Otherwise, you may need to go back and repeat some of the steps in the Gather Requirements phase.

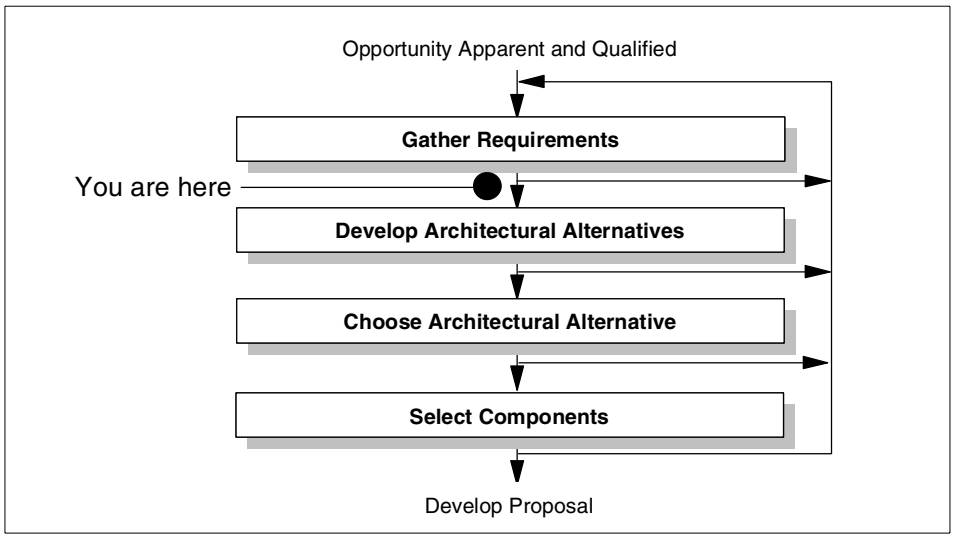


Figure 12. This is Where You Are in the Process

Chapter 4. Develop Architectural Alternatives

In this phase you will start to develop the architecture for your project by coming up with a few alternative architectures from which the final one will be selected.

This chapter builds on the business and technical requirements that you collected in Chapter 3, “Gather Requirements” on page 19. It takes you through the process of developing an architectural model based on the e-business building block diagram shown in Figure 1 on page 1, and provides information which gives guidance on the various choices. It also recommends detailed questions that you should ask and decisions that you should make in order to develop an architecture.

Important

We cannot offer you a simple formula which you can use to develop an architecture based on the requirements. No simple “if requirement X, then architecture Y” statements can be made. Therefore, a large part of this process requires you to rely on your experience, or the experience of people involved in the project, to develop the architecture.

To build on the past experiences of other IBM solution architects, a collection of solution designs, known as intellectual capital, is available for you to use. This subject is covered in 4.5, “Using IBM’s Intellectual Capital” on page 73, and in more detail in Appendix C, “Sample e-business Solution Designs” on page 147.

Your customer may already have some designs in mind and may even have expressed a preference for certain products. You need to take that into account. However, practice has shown that it can be helpful to review architectural options at this stage, and to do so, as far as possible, without making assumptions about specific product choices.

The benefit of this approach should be an increased confidence on the part of the customer in the solution that is finally selected, whether it be the originally envisaged architecture or whether the exercise has highlighted useful alternatives.

Chapter 5, “Choose Architectural Alternative” on page 81, provides guidance on choosing among the architectural options.

If the e-business system that you are designing is particularly large or complex, you should consider involving a practitioner trained in one of the architecture design methods, possibly as part of a services contract.

Appendix F, “Other Design Methods and Sources of Information” on page 163 describes other IBM methods and tools which can be applied to e-business projects.

4.1 Overview of the Developing Architectural Alternatives Phase

The main input to this phase of the design process is the list which you drew up following the steps in Chapter 3, “Gather Requirements” on page 19.

Appendix F, “Other Design Methods and Sources of Information” on page 163 tells you where to find intellectual capital and other reference material within IBM.

Figure 13 shows the steps in the architecture development phase of the e-business solution design.

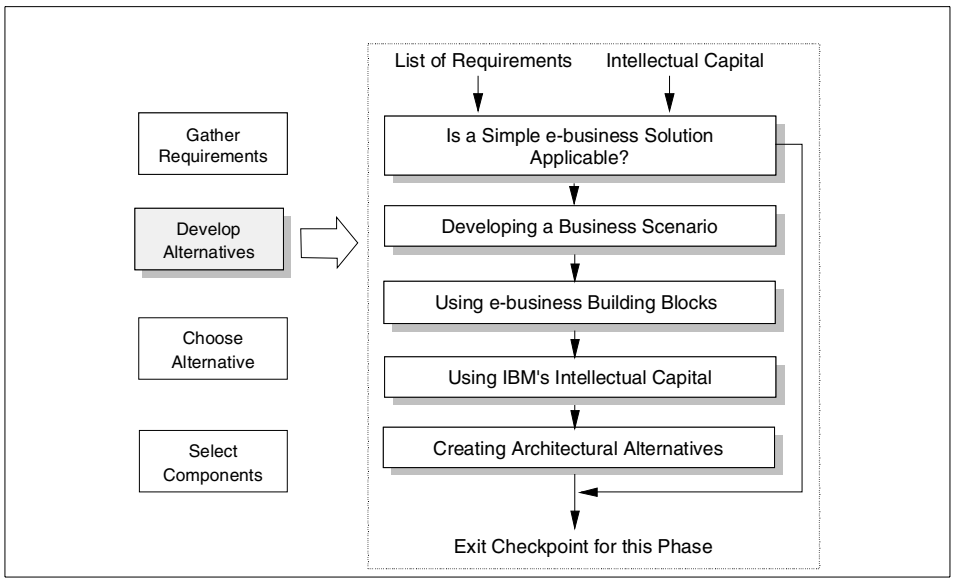


Figure 13. Overview of the Developing Architectural Alternatives Phase

After an initial check whether your requirements are so clear-cut that you can base your solution on an existing design, you create a business scenario (used as a requirements-based guide and tool to validate the alternatives),

define some fundamental aspects of the architecture, and then develop architectural options, building block by building block.

It is helpful to construct the architectural alternatives in a workshop, just as you did during the requirements-gathering phase. Conducting this phase in a workshop setting allows you and the customer to arrive at agreed-upon diagrams of possible solution architectures in a very effective way.

4.2 Is a Simple e-business Solution Applicable

Before investing the time to define your architectural alternatives, you should briefly review the requirements to see whether a clear, agreed solution exists. For example, a requirement for a collaborative workflow system might dictate a Notes/Domino solution, or the requirements may be a close fit with the functions provided by Net.Commerce. If all parties to the project agree, you could bypass the step of building architectural alternatives.

Note that even a simple e-business solution comprised of a single component like Notes/Domino or Net.Commerce requires planning for implementation (although that topic is outside the scope of this redbook). However, even if you decide to bypass the formal construction of architectural alternatives, it is still a good idea to review the questions and comments in 4.4.2, “System Management” on page 49 up to and including 4.4.6.1, “Tools” on page 66.

4.3 Developing a Business Scenario

A *business scenario* is a textual description of the problem being addressed. A business scenario is useful for putting the requirements in context, and for proving the high-level designs which you will prepare for use in this phase. The preparation and use of a business scenario is a common practice in consulting engagements.

Is the development of a business scenario a required step? No. Can it be helpful in gaining commitment to your design? Yes. Therefore, we recommend that you employ this tool as a part of the developing architectural alternatives phase.

It would be helpful to introduce and define three terms at this point: a *business flow*, a *walkthrough* and an *event flow*. These are different forms of business scenarios, from very high level to a more detailed description.

- A *business flow* is a high-level description of the ultimate business uses of the solution being designed. These would typically relate closely to the

requirements gathered, and describe what business tasks are targeted for inclusion in the solution.

- A *walkthrough* is a narrative that describes what people and systems will do with the solution once it is implemented. Remember that we are dealing with very high-level designs at this point, so the descriptions will be in terms of activities, and not products or specific program tasks.
- An *event flow* is a more refined form of a walkthrough. It attempts to describe, in high-level terms, the *events* that occur and processes that are invoked because of those events. We do not illustrate an event flow in this redbook.

Which of these you choose to use really depends on which you are most comfortable with, and which you feel will be most effective with your customer.

During the development of architectural alternatives, this business scenario is used to validate the alternatives to insure they are capable of providing for the needs of the business. The business scenario is also useful for confirming the solution architecture that is ultimately chosen.

An example of a business scenario for an electric utility company that seeks to deploy an e-business solution is shown in Table 5.

Table 5. Business Scenario - Electric Utility Company

Business Flow	Walkthrough
Determine replacement costs of raw materials. This will involve accessing outside information sources to determine the current price of the raw material (gas, oil). It will also involve access to raw material inventory to determine how much replacement material will be required.	The production planner, using a workstation, launches an application to determine the cost of raw materials.
Determine the burn efficiency of fuel.	A request is sent to a server, which in turn sends requests to retrieve inventory and forecast information, and to a Web sources (for example, Reuters) to get current raw material prices.
Determine current operating costs. This will involve access to the financial systems and possibly the work management system.	Based upon the returned information, the replacement cost of raw materials is calculated, complemented by the burn efficiency and current operating costs.

Business Flow	Walkthrough
Calculate costs based on a predetermined formula.	Finally, the energy cost is calculated and displayed to the production planner.
Perform market analysis. This will require access to external information sources.	The production planner launches a request to retrieve market information from the brokers' Web server and grid operator's Web server via the Internet.
Create product price based on predetermined profit margins.	The production planner calculates the product price using the daily margin retrieved from the production application.
Contact grid operator and place bid.	The production planner forwards the bid price for the energy to the grid operator via the Internet.

Additional architectural options may emerge as you explore the e-business building blocks and the related technologies found in this chapter.

Because e-business solutions tend to have object oriented components involved (Java, Enterprise JavaBeans, and so forth), you might want to utilize some object oriented techniques during the phases discussed in this redbook. Presenting object oriented analysis and design approaches is beyond the scope of this redbook. However, you are encouraged to become familiar with these techniques (see Appendix H, "Related Publications" on page 173). For example, developing use cases is an effective way object designers have for generating requirements for an application.

While the two columns in Table 5 could represent use cases at different levels, typically, a use case will be documented utilizing recognized use case templates. Use cases tend to be precise, formal, and accountable. What is meant by *accountable* is that a use case document will act as a contract between users and the developers.

Use cases can be developed at multiple levels, usually using a template to enforce discipline in the process. So, if someone on your team is familiar with use cases, you can substitute these for the business scenario step.

4.4 Using e-business Building Blocks

At this point you should have gained agreement from your customer on two key points:

1. The complexity of the requirements does not lend itself to a simple e-business solution, and a more extensive design process (the subject of this chapter) is required.
2. The business scenario created and presented correctly reflects what the customer is trying to achieve.

As stated earlier, you should conduct this phase in a workshop and make good use of a drawing board for diagrams. Depending on the size and complexity of the project, you may or may not run this workshop as a continuation of the process described in “Conducting a Workshop” on page 34. Again, depending on the nature of your project, you may be working with the client, or the IBM team may be preparing a design for presentation to the client.

Building Block Decisions

In the sections on each building block which follow, you will find a table listing Decision Points together with some notes about the impact of each decision on the design of that building block.

Where appropriate, we have included a second table with more detailed information and usage comments about aspects of that building block.

If, because of the resource available and the size of the project, you are not able to answer all the design questions, you should make it clear what assumptions have been made and revisit those questions when you proceed to more detailed design.

Before starting on the individual building blocks of the diagram shown in Figure 14 on page 44, you should cover the two layers in the diagram which overlay the whole design: Security and System Management. The decisions which you take on these two areas will affect every component of the design.

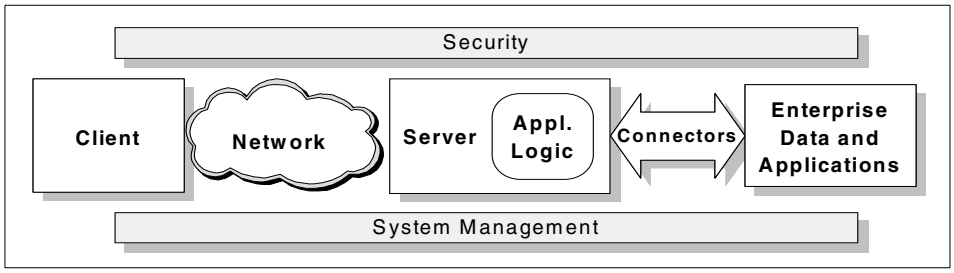


Figure 14. e-business Building Blocks

You should note that although this diagram shows a client communicating with a server, the diagram can, and when applicable should, be extended to cover server-to-server communications.

4.4.1 Security

Table 6, lists questions to consider when deciding on the security policy, procedures and practices for your customer's site:

Table 6. Security Decision Points

Decision Points	Impact
Do transactions need to be encrypted?	Encryption is an expensive operation and will impact your end-to-end budget. Deciding such things as which transactions or which parts of a message need encryption, and the levels of encryption needed inside the firewall versus outside the firewall, may impact the design and certainly the technologies. Technologies such as Secure Socket Layer (SSL) and SET should be analyzed for applicability.
If servers or clients will be located outside the USA, is 40-bit encryption acceptable?	The US is relaxing some controls over the export of cryptographic products. Financial institutions in many countries can now obtain export licences. But if this does not apply, your customer may need to seek strong cryptography from non-US sources.
How will users be identified?	Decide what information will be open to all visitors and at what point users need to be identify. Table 7 summarizes the technologies used to identify users. Cookies can also be used to recognize repeat visitors to the site without needing to know any of their personal details. Use of digital signatures should be evaluated for applicability.
Is there an existing customer database that should be used to identify online visitors to the site?	User profiling can help collect information about visitors to your customer's site providing capabilities to dynamically control interactions with users. Investigating technologies to enable these possibilities should be pursued.

Decision Points	Impact
Does your customer need to restrict access to parts of the site?	The number and placement of firewalls is important to restrict unwarranted access to sensitive data and applications. Reviewing existing IBM Intellectual Assets can help understand proven techniques for using firewalls and designing solutions that properly position components around firewalls for asset protection.
What privacy rules should be applied to information provided by users?	Policies should be in place for the use of collected user data. Many countries now have legislation governing the holding of personal data on computer files.
What are the legal requirements and company policies for auditing content, changes and transactions?	Companies have been held legally liable for the contents of their Web site.
Does the company already have a secure DeMilitarized Zone (DMZ) into which the Web server could be placed?	If so, you must insure that the current Web architecture will support requirements of the new e-business solution, or else decide on compromises to allow your e-business solution to operate within established guidelines.

Table 7, lists various methods of user authentication on the Web, with explanatory notes. Table 15 gives guidance on state management and its security implications.

These security considerations will have to be revisited when dealing with each building block.

Table 7. User Identification

Identification Method	Uses	Comments
Basic authentication	Simple userid/ password authentication	*Basic Authentication* has a specific meaning within the Web protocol. It is supported by most servers - including Domino - and allows simple control over access to files and directories based on a userid. There are two main shortcomings to this method: • The password is masked rather than encrypted for transmission, so this mechanism is not recommended for high-value transactions without some other form of authentication. • The browser caches the password and resubmits it on demand for later transactions. There is no explicit logoff mechanism. This makes it unsuitable for walk up-and-use kiosk applications. Unless a user remembers to close the browser, his identity can be assumed by the next user of the workstation.
Client certificates	Strong authentication	This method authenticates a user with public/private key technology. The technique is used both by Notes/Domino and by the standard X.509 certificates of the Web SSL3 protocol. Because of the infrastructure required to issue and manage certificates, this technology is not yet widespread on the Web, although it is well-handled by Notes/Domino. IBM and Lotus are leading industry initiatives to enhance the interoperability of digital certificates.

Identification Method	Uses	Comments
Application-specific	General user authentication	In this method, user identification is handled completely within the application. The application controls password verification and maintenance. This is the most common way of authenticating visitors to a protected or e-business site. Frequently the application includes a facility for a forgotten password to be mailed to the e-mail address associated with a user's account.
Cookies	Not for authentication, but to associate a visitor to your site with previous transactions	Although it is reasonable to use a cookie to maintain state during a shopping transaction, application designers should not rely on cookies for recognition of users across visits separated by days or more. If the user accesses your site from a different workstation or has reinstalled his software, the cookie will not be available to the server.
Note: The *cookie* is a mechanism introduced by Netscape, but is now standard across most browsers. The server can request that a piece of information be stored on the client computer, which can then be retrieved on a subsequent transaction. Expiry periods can be associated with cookies. For a shopping cart application, the expiry period might be in the order of minutes, whereas for tracking browsing patterns, the expiry period might be months or years. The server matches a returned cookie - which is typically a large random-looking number - against previously issued cookies to determine whether this is a new or a known user.		

4.4.1.1 Confidentiality, Integrity and Accountability

A detailed treatment of encryption tools and techniques and how they can be used to ensure confidentiality and accountability (non-repudiation) is outside the scope of this book. For a reading list together with a basic introduction, see: http://www.ibm.com/security/html/cryptography.html

4.4.1.2 Protecting your Network using Firewalls and Routers

A detailed discussion of the use of firewalls, filters and routes for network protection is outside the scope of this book. You can see an example of a design using firewalls in Figure 43 on page 152.

For more information on all aspects of confidentiality and security start at the IBM SecureWay home page:

http://www.ibm.com/security

Also consult the Security ICM database. See Appendix F.2, “ICM AssetWeb” on page 165 for advice on gaining access to ICM.

Important

Security is a primary area of concern among customers considering e-business solutions. If your customer’s security requirements are very complex, then you should consider involving an IBM Security Architect.

4.4.2 System Management

System Management is a very broad term, encompassing many disciplines, such as performance measurement, availability reporting, reactive problem reporting, software distribution and updates, proactive monitoring to anticipate problems and other problem determination tools. Many of these system management disciplines can be applied, to some degree, even when the network is the Internet (an environment over which you have no control).

However, you should note that some things, such as end-to-end performance management or problem determination of network outages cannot be done for the Internet. Other things, like distribution of software updates, are quite applicable to the Internet.

It is tempting to create a pilot site as a proof of concept and then put the site live on the Internet without considering system management issues. However, it is usually harder to add system management tools and techniques after your site has gone live.

Table 8 shows some of the system management decisions that you should resolve during the architecture development phase and also describes the impact of those decisions on the design.

Note: This table covers policy decisions that you need to take about the management and operation of the system as a whole.

Table 8. System Management Decision Points

Decision Points	Impact
Does the company have the infrastructure to install and run its own e-business server?	If not, you may need to factor into your design ways to develop the appropriate infrastructure. This may lead to recommendations for personnel, technologies and support products, or the customer may wish to have a third party host the service.

Decision Points	Impact
What hours should the service be available?	Bear in mind that the customer's site may be accessed from different time zones and that typical Internet users expect 100% availability.
Is is acceptable to have any scheduled downtime for maintenance?	If not, provision should be made in the design for how maintenance and upgrades are to be made. Availability issues such as these will affect the cost of the solution. If back-end systems are subject to interruption for backup or maintenance, the customer may still want their home page to be available to give details of service status.
How important is it that the service be never interrupted, even for unscheduled component failures?	Avoiding all single points of failure may be costly. Considerations for redundancy in all components should be investigated.
If interruptions do occur, what should be the target time for restoring service?	This implies that service level agreements should be established. Knowing the requirements for the service level agreements will impact the decision points for availability of your e-business solution.
How should partial or total service failures be monitored and handled?	Think about how the application will handle failures. Technologies such as Simple Network Management Protocol (SNMP) can assist in monitoring the application environment by making use of Management Information Base (MIB) standards. You should investigate how well the management tools you select support these management standards.
What are the response time targets?	Response time targets will probably vary by transaction type. Establishing an end-to-end budget for response time provides a way to checkpoint each component within the building blocks during the design phase, and also later during implementation and testing.
Do you need a recovery plan for this e-business system or will it be covered by your organization's existing processes?	This should cover loss of individual components such as hard drives or processors, as well as a disaster which involves the entire site.
How should the architecture support the process of problem reporting, tracking and fixing?	Who should be alerted, and how, if there is a hardware or software problem that affects the site?

Decision Points	Impact
What statistics do you need to keep about the site, and how will they be analyzed?	One essential measure of the effectiveness of a site is the number of visitors. It is also important to understand how easy it is to find information by tracking their navigation patterns.
What instrumentation should be included in the design to measure performance, response times and availability?	You need these details to make informed decisions about capacity plans. Also, your application can provide event information so that operations can take proactive steps to recover in case of an application failure.
Should the architecture include a repository for statistical data?	Can you make an estimate of the storage space required by the repository? Does the design need to include off-loading of historic statistical data to archival storage?

4.4.3 Client Building Block

Table 9 lists important questions to ask when defining the architecture for your Client Building Block.

Table 9. Client Decision Points

Decision Point	Impact
Who is the customer (Internet or intranet)?	On an intranet you may have some control over the hardware and software specifications of the clients. You may even choose to use some client types other than, or in addition to, a standard browser. If your application is to be made available over the Internet, your design point will probably be the two leading browsers.
What is the level of the user's skill?	This will impact screen design. You may wish to plan for Human Factors studies of the dialogs. Other related factors include possible requirements to support text-only mode of operations for sight-impaired people or for scientific professionals who have been reported to prefer the speed of text-only mode when reading academic papers online.

Decision Point	Impact
What languages should the site support?	You clearly need to support the language of the country of your e-business customer set. If you need to support more than one language, you should think carefully about the implications of maintaining versions of your pages in different languages.
What are the user's usage patterns?	Recent studies show two distinct patterns of usage of the Internet to find information: browsing and direct searching. You should consider making it easy for visitors to your site to use either mode. Human Factors also recommends that you should be able to find any piece of information within three clicks. Additionally, your business requirements may require transaction capability for your users.
Is there a need to distribute application code, and if so, how will if be done?	Even if you are just making your application available over the Internet, you may not be able to assume that your potential users have Internet accounts. You may need to distribute guidance and even code to help them access the Internet. If there is application code other than the browser which cannot be downloaded and needs to be physically distributed, you will need a support structure to handle this.
How will the application maintain state?	See Table 15.
How will the choice of client affect end-to-end response?	You should consider key Internet technologies and how they might affect response time. • HTML is the common way to render pages on a browser. Design of HTML pages is an important aspect of end-to-end response time. For example, if your page has many complex objects, the number of connections required by HTTP may be large, download time may be impacted, and rendering time on the browser workstation may suffer. • The use of Java applets will slow down initial response while the applet loads, but may allow subsequent interactions to be faster. • Technologies such as IIOP and RMI should be investigated for applicability. Decisions here may be influenced by desire for open standards.

Decision Point	Impact
Is the browser the only user interface (UI)?	There may be requirements for mail or collaborative applications for which a browser is not the best (or only) option.

Table 10 provides background information to assist your choice of client architecture. It lists the most common types of client that are found on the Internet, together with comments about their applicability to e-business solutions.

Table 10. Client Architecture

Client Type	Uses	Comments
Standard browser	You must design for a standard browser when you cannot dictate the choice of client, for example when dealing with the general public or with third parties over the Internet.	Avoid designing for a specific browser. One British supermarket implemented an online store which was specific to Internet Explorer and rejected users with Netscape Navigator. Following negative publicity, the site was redesigned to support both leading browsers. Even within a standard browser, you need to determine your design point: 640x480 with 256 colors or above?
Specific browser	Preferably never.	Even for intranets where you can dictate the choice of client, the best practice is to support multiple leading browsers. This preserves your ability to extend the application to third parties or newly acquired subsidiaries who might have different browser standards.

Client Type	Uses	Comments
Standard mail client (POP3 or IMAP)	Informal communications.	Because these mail standards are so widespread, it is important that any company wishing to establish a Web presence provide a way to handle enquiries submitted by e-mail. However, as there is no way to be sure that the originator of an incoming e-mail is who his signature says he is, you should not use e-mail for value transactions without some added form of authentication.
Secure mail client	For e-mail with encryption and strong authentication.	This can solve the e-mail requirements for confidentiality, positive identification and non-repudiation. However, there are competing standards (S/MIME, PGP and Notes). As with SSL3 Web client authentication, a certificate management infrastructure is needed.
Collaborative	Applications involving collaboration and/or workflow or requiring strong authentication.	Although mainly used within single-company intranets, there are examples of successful business-to-business collaboration projects.
Thin client	Chiefly used within intranets to reduce cost of ownership. NCs may be used for Web and e-business applications from kiosks to simplify maintenance.	There are several flavors of thin client: Network Computers such as the IBM Network Station; Network PCs; and clients using IBM WorkSpace On-Demand. From the e-business application perspective, treat them as a client with browser and e-mail capability.

4.4.4 Network Building Block

Table 11 lists decisions that you should take for the Network Building Block and some implications of those decisions.

Table 11. Network Decision Points

Decision Points	Impact
Will my solution involve the Internet?	If this is the first time that company systems will be connected to the Internet, you should involve network design and security people from both the client and IBM to agree on policies and design.
What protocols will I use?	HTTP is the standard protocol used by browsers. Other protocols that you might find are FTP (for file transfer) and HTTPS (secure HTTP). Application protocols might include IIOP, RMI, Messaging, or RPCs. The firewalls and filters protecting your site will need to be set up to allow the appropriate protocols to pass through.
What about data, object and application placement?	In order to understand the impact of the e-business application on your network, you need to know the data and application placement. Analysis of projected transaction volumes, amount of data typically being requested, and amount of object interaction is required to help with decisions on your architectural alternatives.
What security functions are required/provided by this building block?	Protocols can affect security design. You need to check whether IIOP, RMI, Messaging or RPCs that you may be considering can flow across your network. The level of encryption that you are considering will affect your network requirements and will have performance implications.
Will my existing network function as required?	While my current network infrastructure may function with introduction of a new e-business solution, additional work may be needed in network design. The network may need to be expanded with additional function or capacity. This activity would typically be a separate effort, perhaps conducted in parallel with development of the new e-business solution.
How does the network affect my end-to-end response time?	Compare estimated traffic against available bandwidth to come up with initial estimates of network responses. Take into account peaks as well as average loadings.

Table 12 lists some network technology options with comments about their uses.

Table 12. Network Architecture

Network Type	Uses	Comments
IP - Internet	Links to third parties or general public	It will be impossible to commit to end-to-end response time targets when transactions are routed over the open Internet.
IP - Intranet	By definition, intranets	An intranet can be securely extended across an open network to create a Virtual Private Network (VPN) using, for example, the IBM eNetwork Firewall product. You can use this technology to link different regions of the same company, or a company with its business partners.
IP - LAN	Intranets	LAN (as opposed to WAN) IP connections allow for greater bandwidth applications.
Proprietary	Existing subnetworks	An existing subnetwork using a protocol such as SNA, DECNET or Appletalk may form part of an e-business network. Network-dependent middleware will be needed to connect to back-end systems.

4.4.5 Server Building Block

In Table 13 we list questions to consider when selecting server components.

Table 13. Server Decision Points

Decision Points	Impact
Is this an end-user client to server model, or will remote server-to-server communications be required?	If other existing servers are involved, how does that affect the choice of this server? Understanding the technologies available can help with possible alternatives. • Messaging: Excellent choice if there are multiple servers that the application needs to interact with, but have different operating environments. May require some programming on the target server to accept and reply with a message. • RMI or IIOP would be considered if the applications on the other servers are object based. • Proprietary interfaces (RPCs, for example) may be needed for certain servers. • Standard SQL access might be appropriate for Relational Data Base access. JDBC or ODBC or a native interface may be needed.
Does this design call for a single server or multiple servers?	You may choose to use multiple servers for load balancing and redundancy or to run different kinds of workloads. (IBM separates its internal mail servers from its application servers). If you intend now or in the future to load balance across multiple servers, what are the implications for concurrent database updating? Capabilities of the application server such as caching, clustering, and load balancing should be considered and will have an impact on component selection.
Does my choice of client options affect the server design?	For example, you may have decided that, as the general public is involved, your design point will be a low specification PC and you cannot assume that the client can run Java applets with acceptable performance.

Decision Points	Impact
How will I implement my application server business logic?	Will I use servlets, distributed objects, CGI programs or server APIs? Usage patterns of users, skill level of developers, number of concurrent users, expected response times all are factors that will affect how the business logic should be architected. For example, casual, infrequent access by a user may dictate a simple CGI or server API interface, while a highly interactive user requiring repeated interaction with the server may dictate a servlet or a distributed object approach.
Will I use applet gateways?	An applet gateway runs on the server and interfaces between an applet on the client and the back-end system to which it is the gateway. This can be a straightforward way to Web-enable back-end systems with a minimum of modification, but will also dictate some of your server options.
Do I need to provide mail or conferencing facilities?	You need to understand how the mail or conferencing will be handled and what functions the server will need to provide.
Do I have workflow requirements?	If so, what level of function do you need? Can you provide this using a normal Web development tool or an integrated solution such as Lotus Domino, or do you need a special purpose workflow package?
Should the server provide indexing and searching or other site navigation aids?	Some servers, or server packages, provide these capabilities as standard, but depending on your requirements you may need a special purpose indexing/search engine. List the searching capabilities that you need: Do you need to be able to search a single server or many? What should be indexed: HTML files, word processor files, PDF files, other formats, databases?

Decision Points	Impact
Is distributed object support needed?	Requirements for a highly interactive transactional system may lead to a requirement for distributed objects to provide services across the enterprise, possibly from several servers. You may need to include an Object Request Broker in your architecture to provide distributed object services. Evaluation of technologies for distributed objects such as IIOP and RMI will be needed, plus understanding your customer's requirements for standards such as CORBA.
What security functions are required in this building block?	All areas of security should be considered: confidentiality; integrity; authentication; access control; auditing capabilities. While an in-depth treatment of security is outside the scope of this redbook, see Table 7 for insight into authentication approaches.
How can I estimate the server component of end-to-end response time?	This is usually not straightforward. The Standard Performance Evaluation Corporation at http://www.specbench.org now includes SPECweb and SPECJVM among its published results. At best, these results can be used to gain a general feel for the relative speeds of various processors, and allow a very rough estimate of transaction throughput. In practice, the results that you will achieve will depend as much on your application design. You should include some performance testing in your project plan, such as that described in Appendix B, "Test for Indicators of Performance (TIP)" on page 139.

Table 14, summarizes the various types of servers that are found on the Internet and comments on their applicability to e-business designs.

Table 14. Server Types

Server Type	Uses	Comments
Generic HTTP server	Basic serving of flat HTML files.	Basic HTML serving is not enough for most of today's needs.

Server Type	Uses	Comments
Combined Web and application server	e-business applications	Most Web servers provide additional application support, sometimes bundled with the server and sometimes available separately and callable via interfaces that the server provides. At this stage in the design you should identify the functions that you require from the server. For example, you may need the application server to provide support for: • Servlets • Java Server Pages • Caching • Clustering • Load balancing • State management • User profiling • The ability to build a searchable index
Mail server	Mail services	Although an e-business will wish to provide its own employees with e-mail capabilities, it will not normally want to act as the home e-mail server for people outside the organization. However, you may well receive requests by e-mail and wish to build a workflow application to handle this.
News server	Discussion, conferencing	We identify this as a specific server type because the conferencing (NNTP) protocol is well known on the Internet and is older than HTML browser technology. An e-business may well wish to provide conferencing in support of its products (for example, the Agfa company does this). Because the NNTP protocol does not normally work through companies' firewalls, if you have a conferencing requirement you may choose to use a product which supports conferencing using browser protocols - such as Lotus Domino or Allaire's Cold Fusion.
Special purpose server	Multimedia, real-time	There are special-purpose methods of serving audio and low-bandwidth video to browsers, such as IBM's Bamba. Real-time conferencing solutions include Lotus Sametime.

4.4.5.1 Platform Selection

In this chapter and phase of the design we are trying to develop an architecture independent of specific products. The questions about the server have not been platform-specific. Considerations for platform selection are covered in 6.5.1, “Hardware Platform” on page 125. Nevertheless, if there are overriding reasons for selecting a particular hardware platform, you should note them here.

4.4.5.2 Session and State Management

Arguably one of the reasons for the initial success of the Web was the simplicity of its “stateless” protocol. A server could treat each request as an independent event and did not have the overhead of keeping track of open sessions. This model was fine when the Web was used for retrieving individual pieces of information, but it does not hold good for e-business applications where, for example, a purchase transaction continues over several interactions with the server. To meet this requirement, various techniques were developed, as described in Table 15.

Table 15. Session and State Management Techniques

Type	Uses	Comments
Cookies	Maintain session state	See the note on page 48 in Table 7, 'User Identification' for a description of cookies. Also note that a user may set his browser to refuse cookies because of (unfounded) fears that they introduce privacy exposures. For that reason your application should be designed to work when cookies are unavailable.
Hidden fields or URL component	Maintain session state	Hidden fields or strings appended as options to a URL can be used to contain an encoded field which defines the session state. Note that as these are visible to users either directly or via the browser View Source command they should not contain, for example, passwords in cleartext. The server should also take care to destroy the session variable at the completion of a transaction because the information could persist in the browser’s cache. This would allow a subsequent user of the workstation to attempt to continue the previous user’s transaction.

Type	Uses	Comments
Applet	Manage session	An entire transaction consisting of several interactions with a server can be performed by a single Java applet which therefore maintains state. This is more complex to implement as it involves selecting and implementing a protocol for communicating between the applet and its partner on the host.

4.4.6 Application Logic Building Block

The decisions you will take about application logic are closely linked with those in 4.4.3, “Client Building Block” on page 51.

Table 16 lists some decisions you should take in determining what style of application development to use.

Table 16. Application Logic/Development Decision Points

Decision Points	Impact
Is this a logical two-, three- or n-tier solution?	Note that we are referring to logical tiers and not physical tiers. It is perfectly possible, and sometimes advisable, to design a logical three-tier solution having two of its logical tiers (typically the Web server and the data server) on the same physical server.
Has it been determined whether to use Object Technology, traditional programming development or integrated packages?	Existing packages, if they meet requirements, are usually faster to implement. Object Technology, with its built-in reuse capability, also offers implementation benefits in conjunction with appropriate development tools.
Is application control needed to determine what information the end-user can print or save to disk?	If your customer is selling information, he may wish to make it hard for the user to save or print it, but this also complicates the design.
Is normal HTML presentation adequate, or should the user interface be enhanced through the use of Java Applets?	Java applets can be used to make the presentation more flexible and user friendly, especially in higher volume transaction-based applications. However, they take time to download and may be a concern over dial-up connections.

Decision Points	Impact
Is client-side scripting needed? At what level?	JavaScript, JScript or ECMAScript are languages, interpreted by the browser, which can be used to enhance the user interface and perform some basic input validation. Because of implementation incompatibilities between the different levels of browsers from Netscape and Microsoft, you should plan to test against all likely clients.
Will the site use proprietary scripts, tags or plug-ins?	The use of proprietary features is generally deprecated as it limits the application's portability.
Will the site use Java applets? What are their connectivity requirements?	As part of the standard security built into Java, an applet can only communicate with the server from which it was downloaded. If the applet needs to communicate with another server, this has to be done indirectly via its own server or through the use of digitally signed applets.
How will the application be split between client-side logic and server-side logic?	Assuming that a logical 3-tier architecture will be the basis for developing the architectural alternatives, you should determine how to distribute presentation logic. For example, you can use straight HTML. This implies that all user field validations will be carried out on the server. This might be very appropriate for a decision support application where little input is needed from the user. Creating HTML with JSP or servlets might be most appropriate. However, if interaction is needed from the user, such as in transaction-based applications, can the application support the amount of client-to-server interaction that this might produce? If it is decided that the client should perform these functions, is a scripting language needed or should applets be employed?

Table 17 gives detailed comments about some of the application logic decisions that you will have to take.

Table 17. Application Logic

Application Type	Uses	Comments
Client-side JavaScript	Simple animation and input validation.	Now pervasive on the Web and recommended for making pages attractive. Provides a good way to provide field-level validation on input from a user immediately, without need to incur network overhead. Implementations differ, so be sure to test with multiple browsers.
Java applets (client-side)	Animation and data visualization. Other uses include session maintenance and locally-written security functions such as strong encryption and non-repudiation.	Can be a very powerful tool for data manipulation and visualization at the client. Use with care because of the time needed to download Java applets, especially over dial-up connections.

Application Type	Uses	Comments
Server-side applications	The basic building block for tailoring web pages to the user, whether based on some information about the user's preferences, or querying a database in response to an explicit transactional request.	The following are some ways of providing server-side application support: • CGI is the most common. It has low portability, but wide programming language support. Even with *Fast CGI* enhancements, however, it does not scale well. • SSI: Server Side Includes is a simple way to produce dynamic pages. However, it cannot process FORM data and has low portability. • Proprietary APIs such as ISAPI from Microsoft and NSAPI from Netscape improve performance over CGI at the expense of application portability. • Servlets are very portable and make it easy to share data between servlets; they provide easy session tracking. • JSP: Java Server Pages are similar to Microsoft's Active Server Pages (ASP) and SSI, and have the portability of Java. • A Java application can provide for a robust object-based application. It may take advantage of Object Request Brokers (ORB) that can provide services such as locating objects (directory), handling requests to the operating environment allowing platform independence, and enforcing standards of how distributed objects will communicate.

4.4.6.1 Tools

As part of the application logic definition you should try to decide which attributes you will look for in a set of tools. Table 18 lists questions that you should ask when selecting tools for creating and maintaining the site.

Table 18. Tool Selection Questions

Decision Points	Impact
How will the applications be built: using application generation tools, or prebuilt packages, or a combination?	Most modern sites use a combination of introductory pages of regularly updated static HTML and application pages that are dynamically generated.
Will content authoring tools be used?	As well as being a valuable productivity aid, content authoring tools make it easier to enforce design consistency.
What content management tools are needed to control the site?	You need a sign-off process for pages. There may even be a legal requirement to know what was displayed on a particular date. You may also wish to associate “go-live” and “do not display after” dates with documents. For a small site, manual procedures may be adequate; for a large site, they are unlikely to be.
How much integration is provided by the application development tool?	The degree to which the type of Web development is integrated into a single tool can help provide productivity for a developer.
How will a team development environment be supported?	The development tools should support a team environment providing integrity for application code.
Will the code be developed on one platform and deployed on another?	The development tools should allow development of code on the platform that is most productive for developers, and then assist in deployment to other platforms.

Decision Points	Impact
Should the tools be extendable?	Most tools provide extensions to assist in the development process. Polices and procedures should be developed concerning the use of extensions. Considerations would be: • To what degree does the extensions lock you into a specific vendor? • Does the extension generate code that is non-standard or uses non-standard APIs, making it less portable?

In Table 19 we list comments about the classes of tools you are likely to need to create and maintain your applications and site contents.

Table 19. Tools

Tool Type	Uses	Comments
Application creation	Create business logic on application server	Web application creation tools should provide programmer productivity features, such as visual programming, project-based team development environment, and support for JavaBeans.
Content authoring		Although it is very easy to display a few pages on the Web, you will probably want to treat your company's pages on the Internet just like any other external publication. That is to say, they should be reviewed before publication for content as well as conformance to your company's corporate image, typefaces, colors, page layout. Editorial rules must be defined and enforced.
	Page creation	If not included with your application development toolset, you may wish to use a visual page creation tool such as NetObjects Fusion.
	Graphic design	Graphic design tools are outside the scope of this document. However, the importance of good, consistent and professional page design should not be underestimated.

Tool Type	Uses	Comments
Content management	Content publishing and management, navigation tools	Some content creation tools also support the publication process and provide site navigation aids. If the tools which you select do not provide this support, you may have to devise manual procedures.

4.4.7 Connectors Building Block

Table 20 lists some of the important criteria for selecting connectors.

Table 20. Connectors Decision Points

Decision Points	Impact
What enterprise systems, applications and data does my e-business application need to access?	See 3.5.4, “Understand the Current Environment” on page 29 for a discussion of the considerations. The type of data, application, platform and network may influence the technologies that may need to be considered in different architectural alternatives.
How should data be transferred between different systems?	Relational database systems provide ways of accessing distributed data that handles the data types and data conversion requirements. For other transfer mechanisms, you should determine the complexity level of the data. From this, you should determine if a self-describing message format such as that provided by the XML technology is needed. XML can provide advantages in dealing with different data types. For example, there are reusable components available that will parse XML messages, thus improving programmer productivity.
How current does the information have to be?	If the information does not have to be updated in real-time, you may be able to use local extracts or caches to improve performance.

Decision Points	Impact
Will I require synchronous or asynchronous access?	The use of an asynchronous messaging technology with assured message delivery can simplify the application programming task. The application can continue operation even if the network connection or remote application is unavailable. Asynchronous messaging does not preclude using it in a synchronous mode. Synchronous technology will require a connection to be available whenever the application is available. Because a connection is known to exist, handling commit processing may be easier.
Will I require access to different operating system, network protocols, application environments?	Although TCP/IP is the standard network protocol of the Web, your back-end systems may be using other protocols. Also, applications may be using standard or non-standard APIs. These considerations will help you evaluate the different connector technologies described in Table 21., “Connectors” on page 70. If protocol conversion is needed, you will have to decide where is the best place to do the conversion.
Is a new user interface (UI) required? If so, what kind?	If you are accessing enterprise data or application logic that currently has a defined user interface, a redesign of that interface allowing program access may be needed. Alternatively, the use of technologies allowing access to the logic and data may need to be employed.
Will I require additional security policies?	Enterprise data and applications may have their own access security polices controlled and administered separately from those in place for the Internet environment. If access to enterprise data and applications is needed, how will security policies be enforced? Can the application server act as a proxy to the enterprise data and applications (this implies trust of the application server’s security policies)? If not, you will need to work out how user IDs and passwords are to be mapped from the Web server to the enterprise servers. With this type of information, you need to study how the different connector technologies support security when connecting to enterprise data. Some research with potential products may be needed.

Decision Points	Impact
Can I predict what scalability and performance I will require?	Careful consideration should be taken concerning performance when designing the application. If you choose to go with a prototype (as suggested in Appendix A, “Proof of Concept (POC)” on page 133), you should allow time to review the design for performance before going into production.

As part of the connectors building block you should try to decide which attributes you will look for when interfacing into enterprise data and applications, as listed in Table 21.

Table 21. Connectors

Connector Type	Uses	Comments
Messaging	Asynchronous access to enterprise servers	Messaging middleware can simplify the application programming task by handling queueing, timeout and recovery/restart conditions. You can also use messaging middleware in a pseudo-synchronous mode. Typically, messaging technology can support large message sizes. Some RPC approaches may be limited in message sizes, requiring additional programming to handle large messages.
JDBC/ODBC	Database calls	These are database-independent interfaces for Java servlets or application programs to make calls to databases that may be on the same or another server.
Native interfaces	Database calls	Many database vendors have implemented native application program interfaces to their own databases which offer a performance advantage over ODBC at the expense of application portability.
Remote Procedure Call	To call programs on remote servers	You may not need to program at the RPC level if you have an application builder that takes care of this for you.
Conversational	Little used in e-business applications	Typically low-level program-to-program communication using protocols such as APPC or Sockets.

4.4.8 Enterprise Data and Applications Building Block

It is very likely that the company already has one (or possibly more) databases holding enterprise data. If so, the design and implementation of the enterprise data repository may be predetermined.

Table 22 lists some questions about data access that you should consider in your design.

Table 22. Enterprise Data Access Decision Points

Decision Points	Impact
Do the service hours for the enterprise data repository match the e-business application targets?	You may have to take some policy decisions about how to treat requests which cannot be handled immediately: options include asking the user to resubmit, or placing the request on a queue for handling later, with possible confirmation by e-mail.
How will you map the access authorization rules for the corporate database to your e-business user identification?	This will be an important question if you intend to enable a controlled ability for certain Web users to update corporate database records.
Is your e-business application using the corporate data for reference (that is, read-only access)?	There may be performance or other advantages to working with a local extract of the database if the requirement for data currency permits it.
Is the data in a format easily accessed by distributed systems?	Enterprise Data may be held in EBCDIC and require translation to ASCII for presentation on the browser. Implications which may (or may not) affect your design include different sorting sequences and number formats.
If additional code is needed to gain access to data (for example, non-relational data), how will this code be developed?	If your enterprise data is located on a mainframe, you may need to develop additional CICS transactions or MQ Series Triggered transactions to interface to existing data or application logic. You will need to decide how this code will be developed and if there are resources available to develop this additional interface code.

Decision Points	Impact
If access is to relational databases, can the SQL be structured to minimize network traffic?	When enterprise data can be accessed with SQL calls, you should review the number of calls needed to satisfy a request. Since issuing SQL calls can be simple, it is sometimes tempting to rely on this when designing the architectural alternatives. Be careful that your architecture does not force you into issuing too many SQL calls, thus reducing performance because of network traffic. Consolidating SQL calls with appropriate use of joins can reduce network traffic. Using stored procedures to consolidate SQL calls can also reduce network traffic. Using a transaction monitor can also provide a way of reducing server-to-enterprise server interactions. Also, resist the temptation to design business logic into the stored procedure, because you will lose flexibility in your design.
What are the commit and rollback requirements of the application?	For Web applications you need to pay particular attention to incomplete or cancelled transactions. At times of slow network response, the user is very likely to press the Stop button on the browser and immediately retry the transaction. Your application design must allow for this.

Table 23 gives brief generic comments about ways to access to back-end systems.

Table 23. Enterprise Data and Applications

Data Type	Uses	Comments
Relational DataBase Management System (RDBMS)	Enterprise Data	RDBMSs are now the standard data repository: most products offer one or more options for Web-enablement.
Transactional Systems Monitor	High volume transactional access to enterprise data	Special-purpose transaction monitors such as CICS and ENCINA can often handle larger volumes of transactions than Web servers.

Data Type	Uses	Comments
Packaged Solutions	Standard applications	Your client may have standard packages which need to be included in your solution. Examples include Enterprise Resource Planning (ERP) packages from various vendors or business software from companies such as SAP.

4.5 Using IBM's Intellectual Capital

IBM has participated in the design of hundreds of e-business solutions. This collection of experience is a very valuable resource, and IBM's strategy is to use the results of this experience in future engagements.

If IBM has this collection of known-to-work solutions, why spend the time with this redbook? Because you will find that no two customers are exactly alike, and no two e-business opportunities are exactly alike. At this point in the e-business solution design approach discussed in this book, you should have completed three very important tasks:

- Collected business requirements and other requirements from the customer. Further, you worked to get agreement from the customer that the requirements are correct, complete, and may be used to build the solution.
- Developed a business scenario that represents the business processes which your design will support.
- Started the process of developing several architectural alternatives that might apply to the business situation presented by the customer, as defined by the requirements they have provided you.

You could design the architectural alternatives without utilizing any of this collected experience from other IBM solution designers. However, this may result in overlooking things you had not thought of, but others in IBM had. So it would be best to utilize, wherever possible, this *intellectual capital* of IBM.

Does this imply that the designs found in this collection are ready to use, without modification, for any customer situation you come across? Not at all. They are intended as *patterns* to be used *with modification* for your customer's environment.

How will you know how and where to modify the designs found in this collection? You can determine these through your experience and by using the process described in this redbook.

4.5.1 How to Get Access to Existing Solutions

Appendix F, “Other Design Methods and Sources of Information” on page 163 gives a summary of other methods which can be used to address all or part of an e-business solution. Appendix F.2, “ICM AssetWeb” on page 165 tells you about the Intellectual Capital (ICM) databases and how to request access to them.

4.5.2 IBM’s Intellectual Capital and Our Building Blocks

In Figure 1 on page 1 we introduced the diagram that we have used throughout this redbook to represent some essential building blocks of an e-business solution. That diagram was intentionally simple in its design. Its purpose was to remove some of the complex details of an e-business design so that you could focus on the fundamental concepts.

The e-business designs that you will find in the collection of IBM’s intellectual capital may be more complex, as illustrated by the diagram found in Figure 15 extracted from the Enterprise Solutions Structure (ESS) Knowledge Network ICM database.

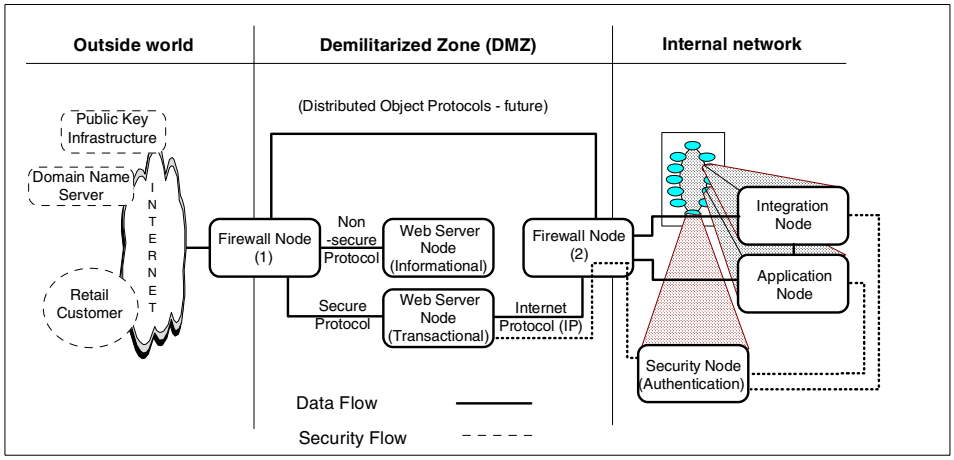


Figure 15. Example of IBM Intellectual Capital Design

As you can see, more detail is provided in this diagram. Now we illustrate how our original building block diagram still applies.

In doing so, we make three points:

1. The *process* of developing a solution involves more than simply taking a previously designed solution and applying it to a new situation. No two

situations are alike and no two customers are alike. *Nothing takes the place of hard work and experience.*

2. The fundamental concept as shown in our building block diagram (Figure 1 on page 1) remains useful as a guiding light through this process of building a solution.
3. *Real* e-business solutions are often complex and rich in detail. You should not be lulled into a false expectation that an easy, quick and simple design will always apply.

First we simplify the design found in Figure 15 on page 74, without reducing its content. This is provided in Figure 16:

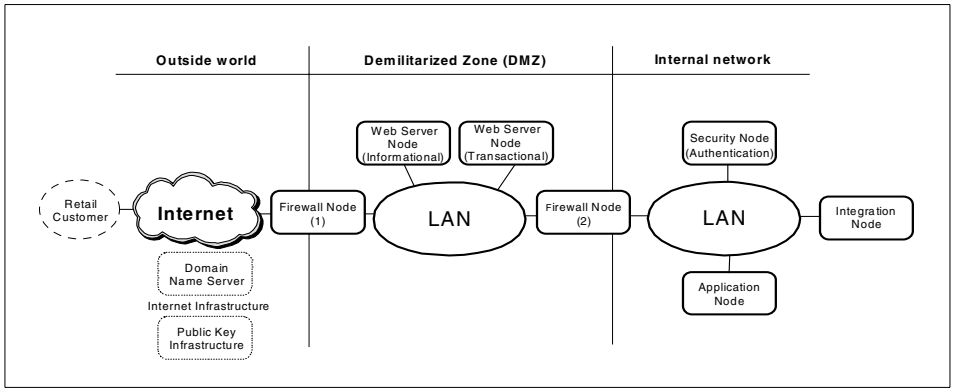


Figure 16. Simplified Version of IBM Intellectual Capital Design

Finally, we map this simplified diagram onto our building block diagram. This is provided in Figure 17:

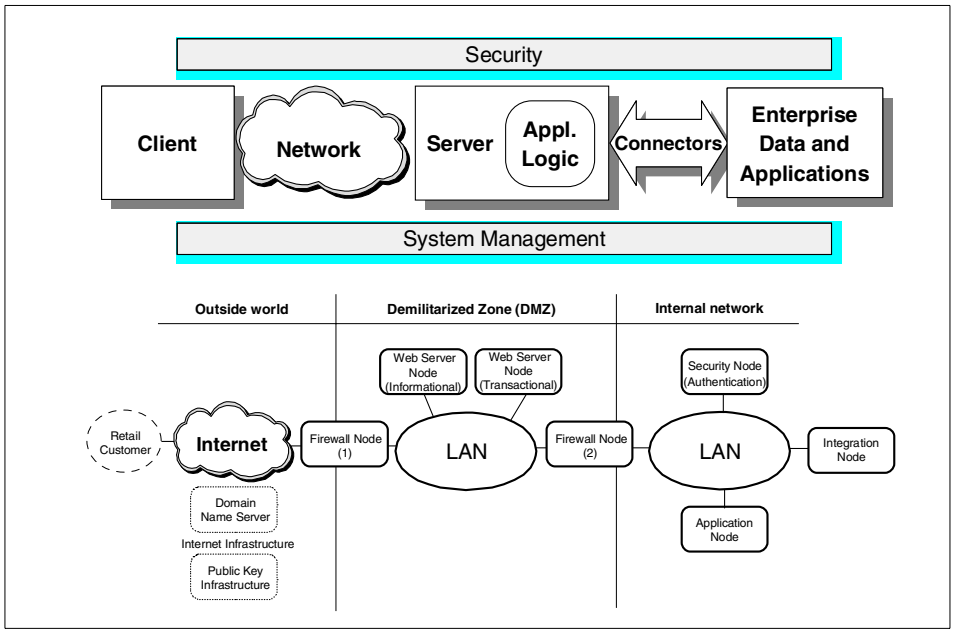


Figure 17. IBM Intellectual Capital Design Mapped to Building Block Diagram

The first thing you will note is that the match is not perfect. That is to be expected. However, as you study this diagram you will see how aspects of the simplified design do map back to our building blocks. This is shown in Table 24.

Table 24. e-business Building Blocks Matched Against ESS Pattern

Object in Building Block Diagram	Objects(s) from IBM Intellectual Capital
Client	Retail Customer
Network	Internet, LAN, and Domain Name Server
Server	Web Server Node (Informational)
Application Logic	Web Server Node (Transactional) and Application Node
Connectors	Integration Node
Enterprise Data and Applications	Application Node

Object in Building Block Diagram	Objects(s) from IBM Intellectual Capital
Security	Firewall (1), Firewall (2), Security Node (Authentication), and Public Key Infrastructure.
System Management	(none shown explicitly)

Note that this and other design assets are accompanied by significant design guidance (for example, in ESS), which itself can be of value in laying out the e-business solution.

More of these sample e-business designs are provided in Appendix C, “Sample e-business Solution Designs” on page 147.

4.6 Creating Architectural Alternatives

Using the requirements gathered in Chapter 3, “Gather Requirements” on page 19 and the information in this chapter, you are now ready to build some architectural alternatives. We recommend that you keep the number of alternatives to a minimum. In practice, three or four alternatives seems to work best. Keep in mind that the architectural alternatives are based on the building blocks we are using throughout this redbook. Your alternatives will be variations of this basic concept. The variations can be as simple as differences in the technologies used. For example, one alternative may have the selection of a CICS connector versus an MQ connector as its only variation. “Example Architectural Alternatives to be Graded” on page 86 gives examples of the architectural options that you might have created.

Prior to beginning this step, you should become familiar with IBM's intellectual assets as described in “Using IBM’s Intellectual Capital” on page 73. These assets can help you determine the direction that the solution for your customer may follow and give you ideas on how to best illustrate the alternatives. You should be able to relate the architectural components of IBM's intellectual assets to the building blocks. An example of this mapping is described in “IBM’s Intellectual Capital and Our Building Blocks” on page 74. Keeping the diagrams simple can help you better position the alternatives and facilitate the discussion with your customer. The key is to document your alternatives clearly so that as you present them to your customer they are easily understood.

To insure your architectural alternatives are capable of satisfying the business problem, you should use your business scenario to validate them.

Once documented and validated, you should present your alternatives to your customer. It is helpful if you develop a short list of pros and cons for each of the alternatives, utilizing your team's knowledge and experience, information gathered from this redbook, and information available from various vendors (references are given throughout this redbook). It is important to keep the discussion at this point focused on the technologies offered by the alternatives. Avoid the temptation to talk about products. The development and discussion of pros and cons can help position your alternatives and is excellent preparation for selecting the best architectural alternative as described in Chapter 5, “Choose Architectural Alternative” on page 81.

4.7 Chapter Checkpoint

At this point you should have created architectural alternatives, documented and validated these alternatives, and presented them to your customer. With the help of your customer, you are now ready to select the best architectural alternative, but before proceeding ask yourself the following questions:

1. Have you utilized IBM Intellectual Assets to help reduce the risk associated with the alternatives?
2. Have you reviewed your alternatives with appropriate specialists or architects?
3. Have you developed and discussed a short list of pros and cons for *each* of the alternatives?
4. Have you clearly documented and validated the architectural alternatives?
5. Have you gained commitment from your customer on the alternatives and technologies presented?
6. Have the appropriate actions items with owners assigned been put into place for any unresolved issues?

If you can answer yes to each question, you should have some architectural alternatives agreed to by your customer that you can now use in Chapter 5, “Choose Architectural Alternative” on page 81, to arrive at the best architecture to meet your customer's requirements.

If you have answered no to any of the questions, or if additional requirements concerning the business problem surfaced during the discussion of the alternatives, you may need to go back and repeat some of the steps in earlier phases, which are shown in Figure 18 on page 79.

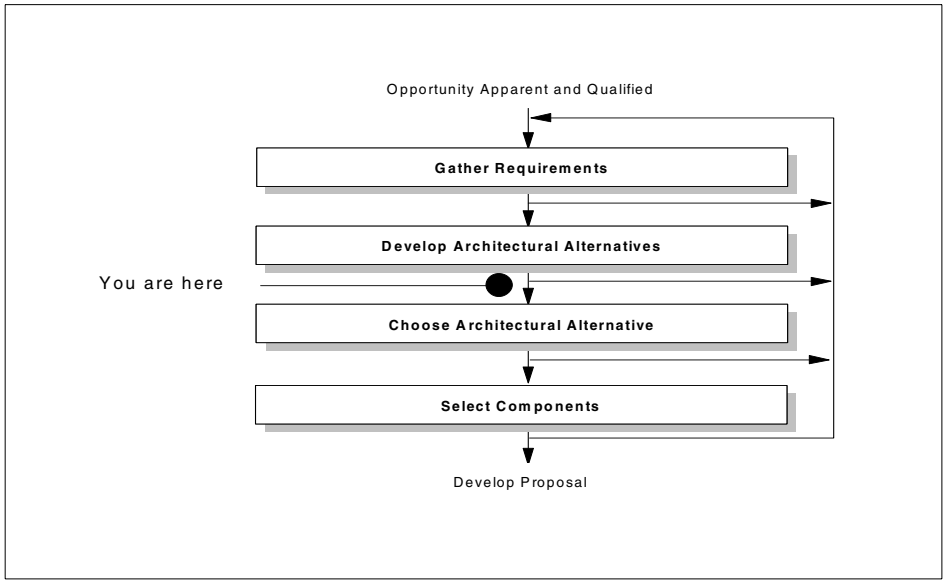


Figure 18. This is Where You Are in the Process

Chapter 5. Choose Architectural Alternative

The purpose of this phase is to arrive at a decision about which alternative to implement. We do this by providing a technique of matching the requirements, determined in Chapter 3, “Gather Requirements” on page 19, to the architectural alternatives derived in Chapter 4, “Develop Architectural Alternatives” on page 39.

It is recommended that this process be conducted in the same workshop environment as used to select architectural alternatives. This keeps the customer involved in the process and insures concurrence with the process and outcome.

The idea is to keep the process simple and to the point. While many techniques could be used to determine the architecture for your e-business solution, the approach given in this phase has proven to be successful time after time.

Experience has shown that if you have reached this phase, very few additional issues will be raised because most issues will have come out of the architectural alternatives and will have been resolved during that phase. However, if additional issues do surface they will need to be resolved. Ideally, these issues should be resolved during the session by technical experts present or, at a minimum, they should be clearly understood and documented. Follow-up should be accomplished as soon as possible. In this case, the final selection may be made with a contingency on the final resolution of an issue.

5.1 Overview of Choose Architectural Alternative Phase

Figure 19 on page 82 illustrates the high-level approach represented in this chapter.

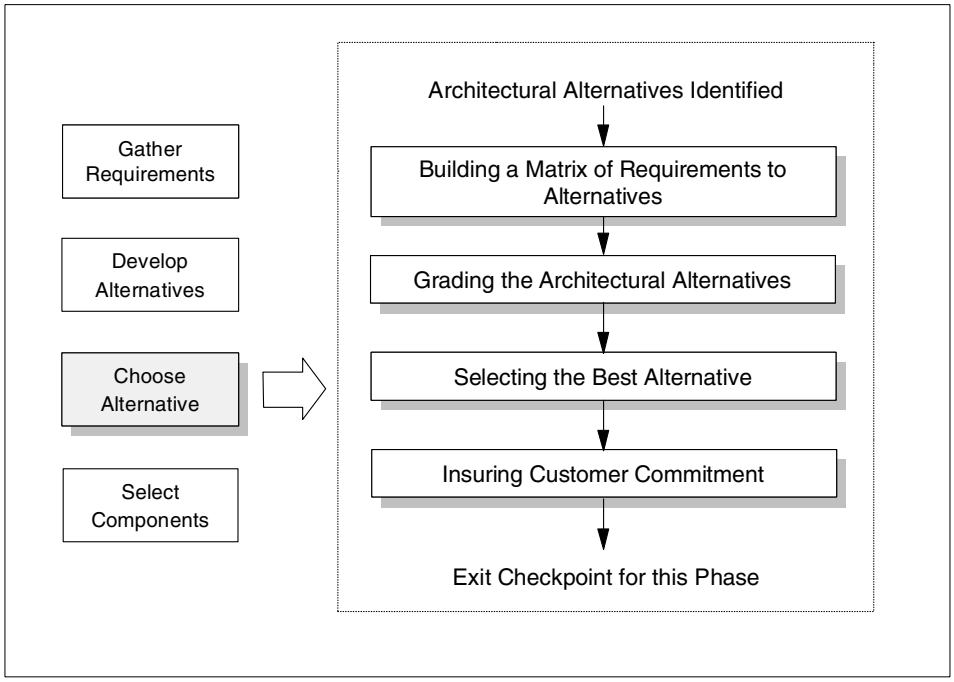


Figure 19. Overview of the Choose Architectural Alternative Phase

5.2 Building a Matrix of Requirements to Alternatives

A matrix that matches high priority requirements (as defined in 3.6.5, “Prioritize Requirements” on page 36) against the architectural alternatives (as derived in Chapter 4, “Develop Architectural Alternatives” on page 39) creates a tool by which an alternative can be chosen. It is important to focus on just the high-priority requirements (and perhaps a few of medium priority), rather than permitting dozens of requirements to obscure the process.

Perhaps an example of a “Requirements to Architectural Alternatives” matrix is the best way to help understand this process. Table 25 shows an example of how this might be presented to the customer. Note that the requirements shown are hypothetical; you would use the requirements list you generated.

Table 25. Requirements to Architectural Alternatives

Requirements	Arch. Alt. 1	Arch. Alt. 2	Arch. Alt. 3
An aggressive production deliverable date			
Maintain or improve performance			
Construct for growth and scalability			
Allow for platform independence (use industry standard interfaces/protocols)			
Lower product ownership costs			
Front-end redesign			
Reduce dependencies on current middleware product sets			
Transfer skills and train customer people			
Position for Internet			

5.3 Grading the Architectural Alternatives

The process of *grading* the alternatives involves making an assessment of how well it meets each requirement. This grading process is best kept simple by using a *plus (+)*, *neutral (o)*, or *minus (-)* symbol for each cell in the matrix.

Table 26 provides a definition of the suggested symbols used during the grading process.

Table 26. Definition of Grading Symbols

Symbol	Description
+	Indicates that the requirement is met and exceeded in some ways
o	Indicates that the requirement is met
-	Indicates that some aspect of the requirement may not be met

Keeping the process simple has proven to be an effective approach. This is why the simple *plus*, *neutral* and *minus* system was recommended in Table 26. Other techniques for grading, such as a weighting scheme (for example, 1 to 10), have been tried. However, the simple approach has proven more

successful. Customers usually can make or accept decisions between +, o, or - without too much trouble. A weighting system tends to generate much more discussion without much additional benefit.

This grading process should be done first by the IBM team, and then with the customer. The reason for this is described in 5.3.2, “IBM Team Preparation” on page 85.

5.3.1 Weighing Alternatives Against Requirements

Your team must look at each of the requirements and determine to what degree each requirement can be achieved by the architectural alternatives. *This process depends on the collective experience of your team in the architectures being considered, and the technologies involved*. The information provided in Chapter 4, “Develop Architectural Alternatives” on page 39 is a basis for making some these assessments. This is a complex arena; seek out skilled technical people to assist.

Table 27 gives examples of how you might weigh the architectural alternatives against the hypothetical requirements we have provided. It shows considerations that must be taken into account.

Table 27. Examples of What to Look for in Architectural Alternatives

Requirements	Then look for an alternative that...
An aggressive production deliverable date	• takes advantage of reusable components. • possesses technologies that are proven. • provides easy integration to enterprise data and applications.
Maintain or improve performance	• meets your end-to-end response time budget (for example, HTTP protocols are not efficient for high volume interactive activity). • does not result in a great deal of network traffic to communicate with a client (remember, you have no control over Internet performance). • will work within the physical limits of the client hardware that is present (a customer with a large installed base of 486 processor machines with 16MB of memory is not a candidate for a large and complex Java application).
Construct for growth and scalability	• scales vertically and horizontally, allowing for maximum growth if needed (you may wish to consider the cost and difficulty to accomplish this scaling).

Requirements	Then look for an alternative that...
Allow for platform independence (use industry standard interfaces/protocols)	• minimizes the use of proprietary technologies. • minimizes the use of non-standard protocols and interfaces. • uses a programming language that will allow the greatest amount of platform portability.
Lower product ownership costs	• allows for easy maintenance. • permits new functions to be easily added without disrupting other functions. • facilitates easy distribution of code to users. • accommodates easy migration from one release to the next. • allows for easy management of operations.
Front-end redesign	• provides sufficient *widgets* available to design the desired user interface (in an e-business solution it is sometimes difficult to match capabilities found in some of the client/server power tools). • has the capacity to provide new capabilities not used or imagined before.
Reduce dependencies on current middleware product sets	• provides an alternate way to access enterprise data and logic without relying on existing middleware to be utilized.
Transfer skills and train customer people	• takes advantage of existing customer skill sets. • has concepts and features that are easier to learn than other alternatives.
Position for Internet	• works equally well over Internet versus intranet (for example, RMI may require special considerations when going through a firewall). • best supports the security requirements.

5.3.2 IBM Team Preparation

It is highly recommended that the IBM Team meet to review the matrix and grade the alternatives before grading the alternatives with the customer present. The advantages and disadvantages of each alternative should be well understood before engaging the customer.

The advantage of this is that it gives the IBM Team an opportunity to understand the positioning of the alternatives relative to the requirements so that whoever leads the customer session can help direct the discussion.

5.3.3 Customer Grading Session

Now comes one of the more exciting phases of the process; trying to narrow the architectural alternatives to one that best meets the customer requirements.

At this point, it might be helpful to review quickly each of the requirements with the customer. It is important that you continue to maintain consensus with the customer throughout this process. One of the most effective ways of accomplishing this is to take every opportunity to review information that has been gathered to make sure everyone is still in agreement.

Likewise, reviewing the architectural alternatives provides an opportunity to insure the customer is familiar with the approach being proposed and understands the technologies being explored. If issues or concerns are raised, they need to be addressed.

Although the grading process should take place rather quickly, opportunity does present itself for discussion of a particular requirement or architectural alternative. While the workshop leader should keep this discussion to a minimum, it does provide the opportunity to further understand the customer's view of a requirement, which may result in re-prioritizing or even eliminating the requirement. This discussion provides an opportunity to *soft-sell* a desired outcome by providing rebuttal to a possible objection, or simply by providing additional information.

5.3.3.1 Example Architectural Alternatives to be Graded

To illustrate the process of grading a set of architectural alternatives, assume you have a customer that wishes to provide *its* customers with a way of accessing enterprise data. Your customer wishes to do this because it will help generate more business. The requirements gathered in the Solution Workshop are those presented in Table 25.

As shown in the following figures, let us say that three alternatives were constructed to address this set of customer requirements.

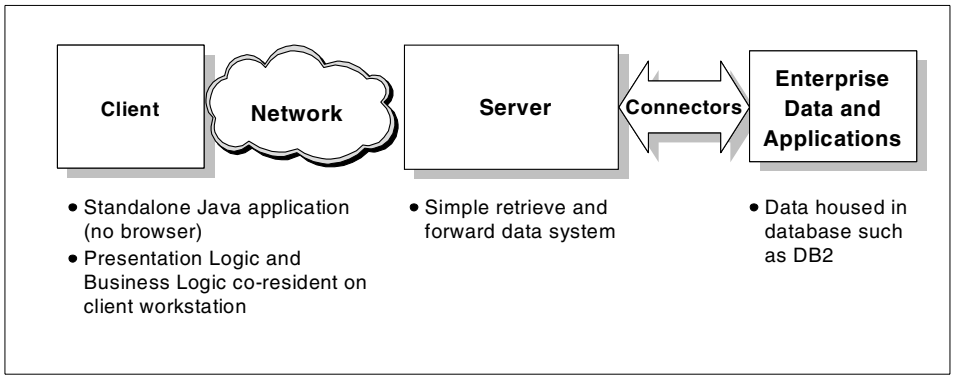


Figure 20. Architectural Alternative 1

In architectural alternative 1, a Java application is developed and installed on each client workstation. This Java application performs all of the business logic as well as presentation logic. All it passes to the server is information on the data it wishes (for example, SQL statements).

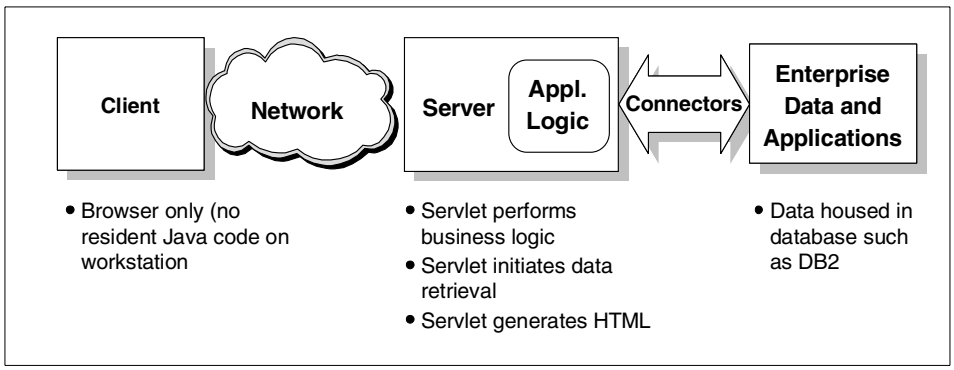


Figure 21. Architectural Alternative 2

In architectural alternative 2, a standard browser is the only additional code added to the client workstation. Servlets on the server perform all the business logic, and the servlet generates the HTML that is sent to the browser. All of the interaction between client and server is performed via HTML.

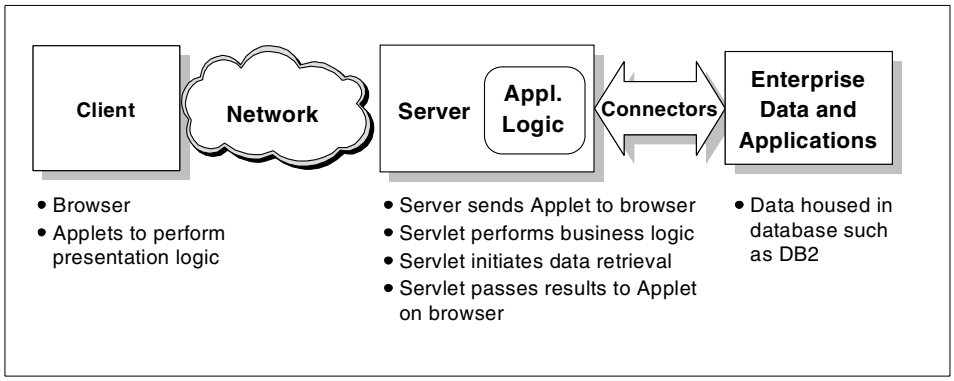


Figure 22. Architectural Alternative 3

Architectural alternative 3 provides a form of hybrid between the two. A browser on the client workstation is used, along with applets that perform the presentation logic. Java Application Code on the server receives data sent from the applet, performs whatever business logic is called for, and then performs the data retrieval. The information is sent back to the applet on the client using the protocol selected for distributed object communication.

5.3.3.2 Example of Grading the Alternatives

Table 28 provides an illustration of this grading process.

Table 28. Customer Grading of Requirements to Architectural Alternatives

Requirements	Arch. Alt. 1	Arch. Alt. 2	Arch. Alt. 3
An aggressive production deliverable date	-	+	o
Maintain or improve performance	+	-	o
Construct for growth and scalability	+	+	+
Allow for platform independence (use industry standard interfaces/protocols)	+	+	+
Lower product ownership costs	-	+	+
Front-end redesign	+	-	+
Reduce dependencies on current middleware product sets	+	+	+
Skills transfer and train customer people	-	o	+
Position for Internet	-	+	+

5.3.3.3 Explanation of the Grades Assigned

To illustrate this process more fully, Table 29 explains the thinking that went into the grades that were assigned to each alternative for each requirement.

Note: This example is based upon a real customer experience. Many of the details of the customer environment have been omitted for the sake of brevity, although such details were used in the design process. When you work with a real customer on an e-business solution, you will factor in hundreds of small details, just as we did on this design.

Table 29. Explanation of Grades Assigned to Example Alternatives

Requirements	Comments
An aggressive production deliverable date	Alternative 1 was graded a minus rating because of the time required to deploy the code to all of the customer site, plus a slight lack of experience in coding Java applications. Alternative 2 received a plus rating because of existing HTML skills, along with the centralized design. Alternative 3 was given a neutral rating because of lack of strong Java programming skills, but the centralized design offset that to produce a neutral score.
Maintain or improve performance	Alternative 1 received a plus rating because a client-side application can be developed to optimize the network traffic and provide a responsive interface. **Note**: Performance in this scenario might be an issue if the workstation is limited in its capacity to run the application. The customer allowed an assumption to be made that the workstations would be of sufficient capacity. Alternative 2 received a minus rating because HTML can prove to be a heavy network protocol, particularly when a feature-rich interface is desired, as was the case with this customer. Alternative 3 received a neutral rating because although it uses a client-side application (an applet), it takes time to download that applet at the start of the session.

Requirements	Comments
Construct for growth and scalability	All three received a plus rating, although alternative 1 might very well have received a minus were it not for the assumption that client workstations will be of sufficient capacity to run the application. Because alternative 1 has its business and presentation logic located in the application, that application could grow quite large. Older workstations, or workstations with limited memory or processing power, might pose a problem as to how far the application on the client could scale. Alternatives 2 and 3 have presentation logic and business logic separated. This permits the server side to scale vertically to the limits of the platform on which it runs, and then horizontally by adding more servers.
Allow for platform independence (use industry standard interfaces/protocols)	All three alternatives employ non-proprietary protocols (IP) and interfaces (Java), so all three received plus ratings.
Lower product ownership costs	Alternative 1 received a minus rating on this simply because of the cost of deploying and maintaining the code on all of the remote workstations. Alternatives 2 and 3, which employ a centralized design, eliminate this concern.
Front-end redesign	The customer's desire was to present a very useful graphical interface. Alternative 2 was given a minus because it is based on HTML. Highly interactive and rich graphical features are difficult to provide with HTML. Therefore, alternatives 1 and 3, which use a client-side application, can achieve better results for this requirement.
Reduce dependencies on current middleware product sets	All three alternatives provide a way to reduce the dependency on the current middleware products. Therefore, all were given a plus rating.

Requirements	Comments
Skills transfer and train customer people	Alternative 1 received a minus rating because the installation of the application on the customer workstations would have required some training, as well as training on how to use the application. Alternative 2 received a neutral rating because although it has the benefit of a centralized design and utilizes an industry standard browser, it uses HTML which restricts the ability to make the interface as intuitive as possible. Alternative 3 received a plus because the interface could be made easy and intuitive due to the applet design. In addition it was a centralized architecture and used the industry standard browser, making updates and installation the easiest.
Position for Internet	Alternative 1 was given a minus rating because other than its use of IP as its transport protocol, it does very little to position the solution for the Internet. The other two alternatives make use of a browser and Java technologies (servlets and applets) that do a better job of meeting this requirement.

5.4 Selecting the Best Alternative

Once the grading exercise has been completed, experience shows that it is rather easy to select a “winner”. The process is not scientific; rather, by the time the customer has graded the alternatives, one alternative tends to suggest itself.

In the example provided in Table 28, the three hypothetical alternatives come out like this: Alternative 1 received five positive marks, alternative 2 received six, and alternative 3 received seven. That would suggest alternative 3 would be the winner.

Sometimes it is useful to weigh the negatives against the positives and come up with a *net* assessment for the alternative. For example, if a negative mark offsets a positive one, and a neutral mark has no value, then alternative 1 ends up with a net +1, alternative 2 with a net +4, and alternative 3 with a net +7. This reinforces the selection of alternative 3.

Finally, though we have not suggested you prioritize requirements within the categories previously described (high, medium and low), it is sometimes useful to go back and determine which requirements, if any, carry the most

weight. For example, the requirement for a rich front-end design might have been the most important factor in the mind of the customer. The negative grade for that on alternative 2 might outweigh several of the other positives.

5.5 Insuring Customer Commitment

Following the approach outlined in this chapter should insure the customer's commitment to the final architecture selection. If there are still outstanding issues that the customer has, these should be listed and specific action items developed to address the particular issue. Commitment to the selected architecture is essential before continuing. If this requires refinement of requirements or of the architectural alternatives, then repeating some previous steps may be necessary.

Perhaps it bears repeating: *designing e-business solutions is an iterative process.*

5.6 Chapter Checkpoint

At this point you should have a selected an architecture for your e-business solution, documented the architecture, and gained customer commitment to the selected architecture.

1. Have the requirements changed by the customer since you have completed this phase? If they have, you must do the appropriate updates and validations
2. Have you completely documented the selection process and agreed-to selection?
3. Have you made any assumptions that have not been agreed to by the customer? If so, you must gain customer agreement on those assumptions. Based on results, you may need to go back to previous phases.
4. Do all open issues have agreed-to owners, action plans, and dates for resolution assigned?
5. Is the customer committed to the selected architecture?

If you can answer yes to each question, then move on to Chapter 6, "Select Components" on page 95; otherwise, you may need to go back and repeat some of the steps in earlier phases.

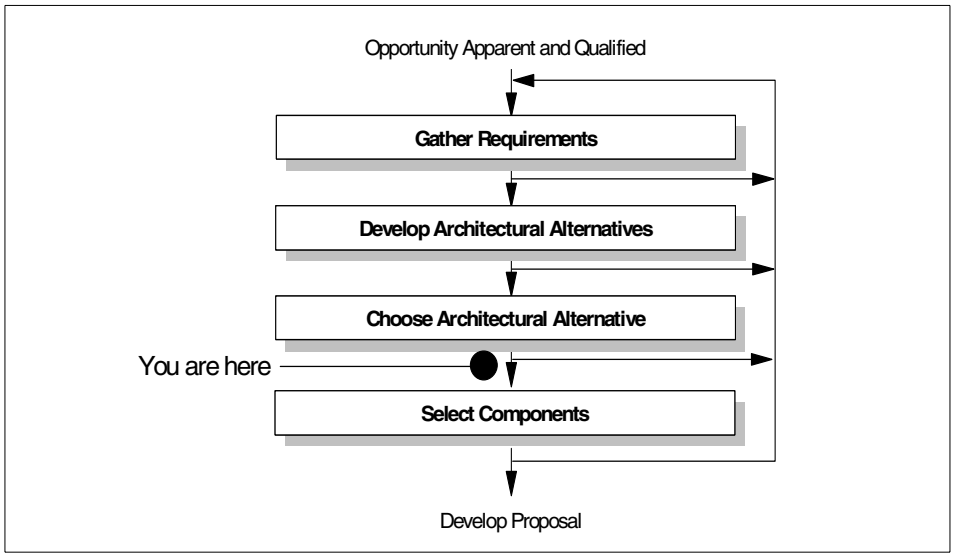


Figure 23. This is Where You Are in the Process

Chapter 6. Select Components

The purpose of this phase is to provide guidance in choosing products to support the selected architecture. It is crucial in this phase to understand the components necessary to support the selected architecture, as well as to understand IBM's strategy for building e-business solutions with IBM products.

This chapter presents a brief overview of IBM's e-business Application Framework (ebAF) and an overview of how some notable competitors position Web services. One approach is to use ebAF as a template for laying out your customer's e-business framework. By doing this, you can help your customer categorize the services they provide, along with how their products map into those services.

ebAF provides a view of Web services and can be used to show how IBM's e-business software portfolio of products relates to the services being provided in an e-business solution. The building blocks used in this redbook can be mapped to ebAF (see Figure 28 on page 100). The building blocks provide a simple way of categorizing the elements of your e-business solution and can help structure your assessment of components needed to support your solution.

Armed with this information, the next step is to understand some of the key products that make up typical e-business solutions. We have used the building blocks as the basis for organizing the discussion of the e-business components. These components, both IBM's and competitors', are listed along with the technologies they support, and a brief description that includes a reference to where more information can be obtained is also provided.

Using the selected architecture determined in Chapter 5, "Choose Architectural Alternative" on page 81, you can select the appropriate components that support your design by making use of the technologies supported column along with the description column, found in Table 31 through Table 37. The intent here is not to provide the last word on components, but rather provide you with guidance and specific references where you can get more detailed information.

6.1 Overview of Select Components Phase

Figure 24 on page 96 illustrates the high-level approach represented in this phase.

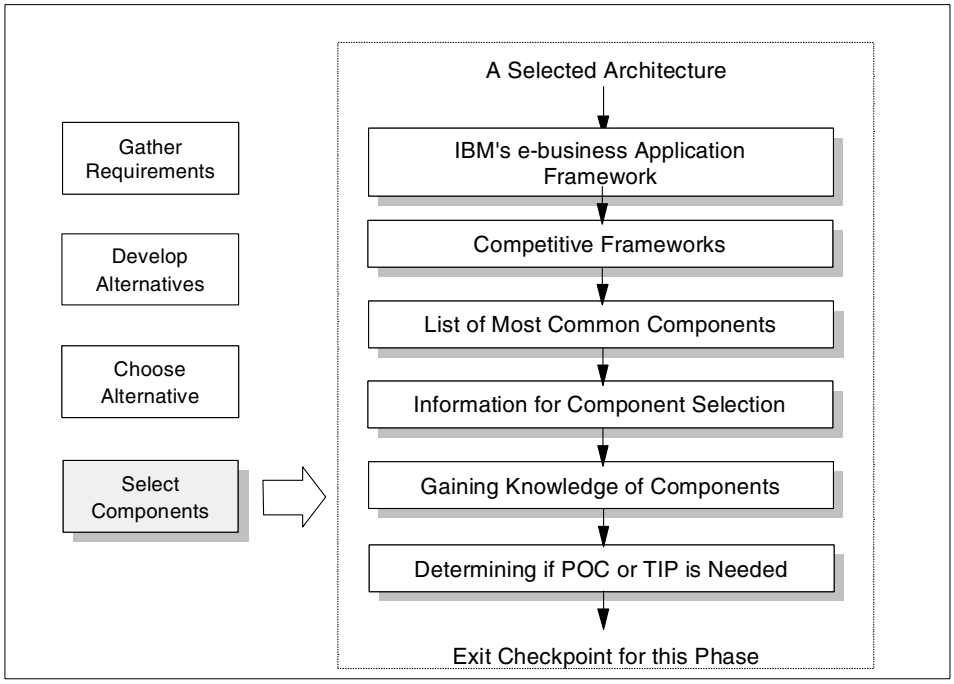


Figure 24. Overview of Select Components Phase

6.2 IBM e-business Application Framework (ebAF)

ebAF is based on a logical three-tier architecture that enables you to extend your existing applications and quickly make the most of your current investments. It enables you to extend, rather than rewrite, your existing applications. This means development times of weeks or months, rather than years.

ebAF also brings Java to the world of business-critical computing, so you can write applications once and run them anywhere, instead of having to create multiple versions for different browsers, clients and servers. Since Java allows you to focus on development, rather than deployment, it gives you the flexibility to put applications together quickly with the capability that they will run later on the platform that makes the most sense. There is no need to synchronize efforts nor to perform rework if the platform changes. Best of all, you will never be restricted to one platform because of high porting costs.

ebAF provides a standards-based infrastructure with a set of common services and a component interface, JavaBeans and Enterprise JavaBeans.

This standards-based infrastructure enables organizations to adopt other third-party solutions as needed. It also enables IBM-based applications built on the framework to interoperate with other standards-based applications. ebAF infrastructure includes:

- Client support
  - HTML, D-HTML, ActiveX controls, Java applets
  - Java Virtual Machine (JVM): provides platform independence for e-business applications
- Standards-based security
  - Security: supports SSL and IPSec for application and network security respectively, and SET for secure commerce
- Standards-based services
  - Directory services: take advantage of the industry-standard LDAP protocols
  - Web server: supports HTTP and IIOP to enable communication with clients
  - Mail and community standards: SMTP, POP3, IMAP4, NNTP, and calendar
  - Database connectivity: SQL and JDBC
  - Transaction services: CICS APIs, OMG Object Services; Open Group's XA reference model standards; JavaSoft Enterprise JavaBeans specification
  - Connectors: JavaBeans and Enterprise JavaBeans
  - System Management

Figure 25 on page 98 shows IBM's e-business Application Framework.

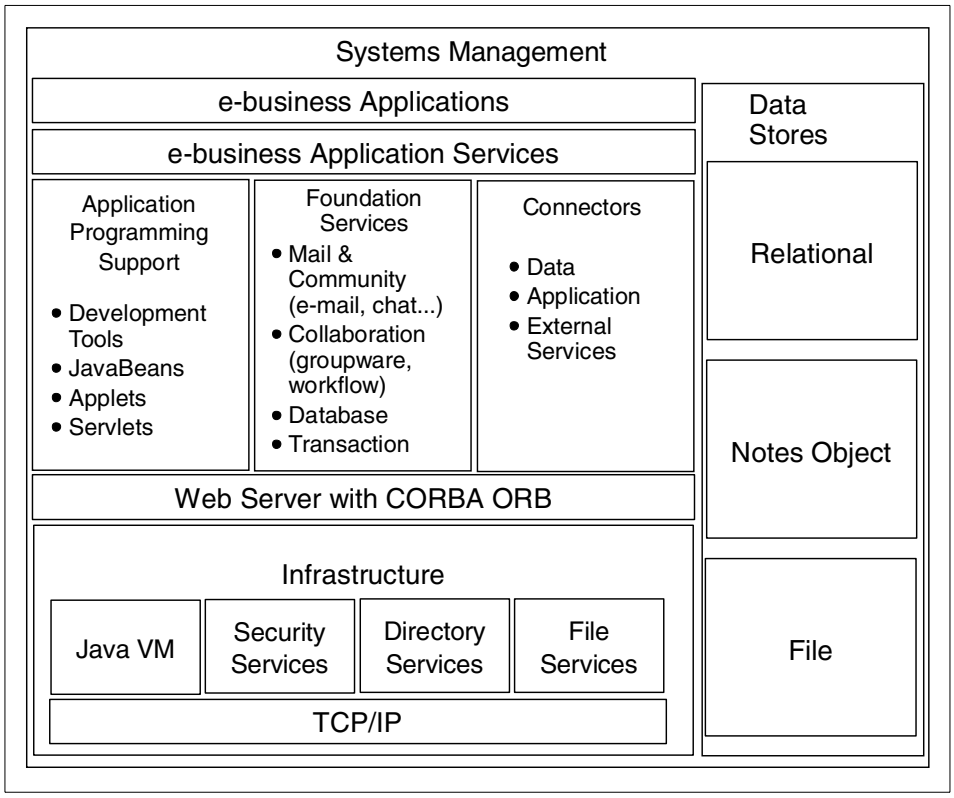


Figure 25. IBM e-business Application Framework

IBM's e-business products can be mapped onto the framework providing a way of categorizing the products as shown in Figure 26 on page 99.

If you use ebAF as a template for laying out your customer's e-business services and map their products on to the framework, you can see where there may be deficiencies in your customer's e-business software portfolio or where they are utilizing competitive offerings. This customer-based e-business framework can be helpful as you progress through the selection of components to support your selected architecture.

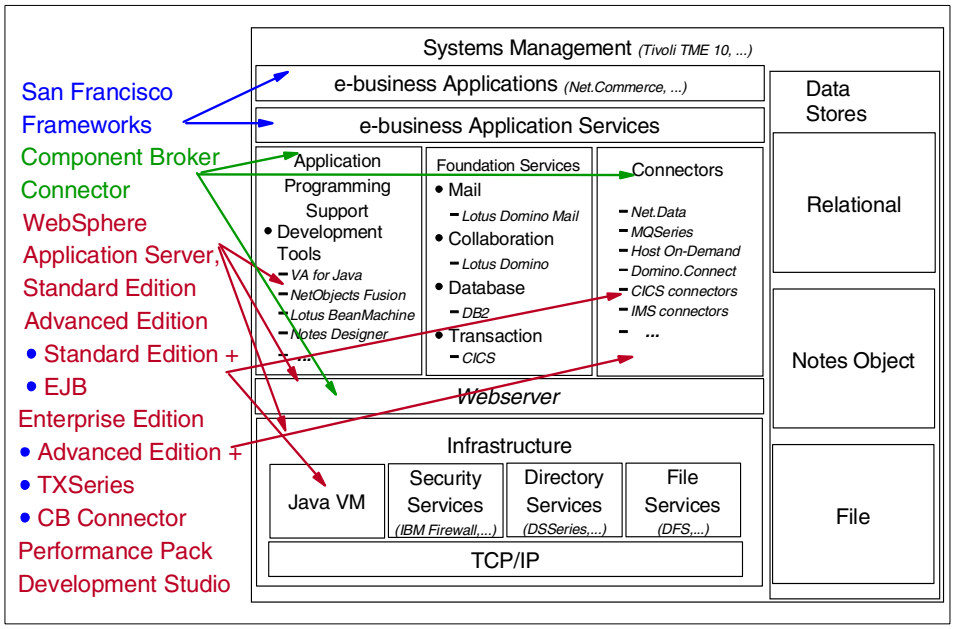


Figure 26. Products Mapped to e-business Application Framework

6.2.1 How the Building Blocks Map to ebAF

Once you have a customer e-business framework you can use your e-business selected architecture (based on the building blocks) and quickly assess your customer's requirements for e-business products. The mapping in Figure 28 on page 100, can assist you in understanding the relationship between the building blocks, Figure 27 on page 99, and IBM's e-business product strategy.

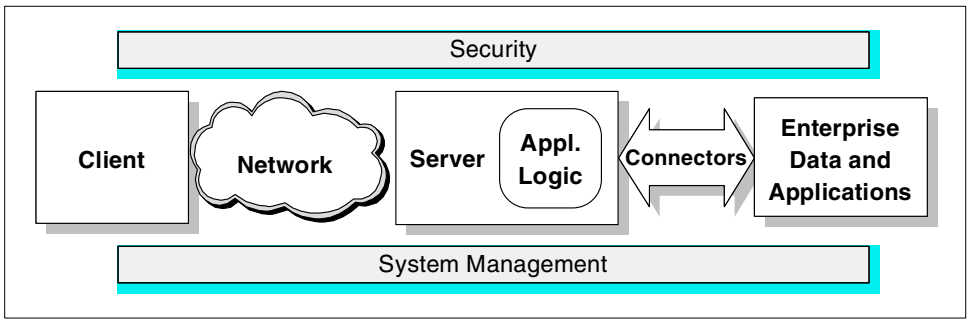


Figure 27. Our e-business Building Blocks

In ebAF, the clients are characterized by a reduced cost of application management. The application elements of a Web application typically reside on a Web application server. Since all HTML pages and Java applets are loaded from the Web application server, no application-specific code has to be installed on the client. The framework clients support industry standard communication protocols such as TCP/IP and HTTP. Additionally, the framework supports application clients (e-mail, chat and so forth) and fully integrated clients such as Lotus Notes. The client portion of the building blocks support these characteristics with the associated products being browsers, Notes client and IBM Network Stations.

The Security layer and System Management layer from the building blocks are not specifically mapped in Figure 28 on page 100, but are represented within the framework as System Management and Security Services. For more details on ebAF, visit: http://www.software.ibm.com/ebusiness

.

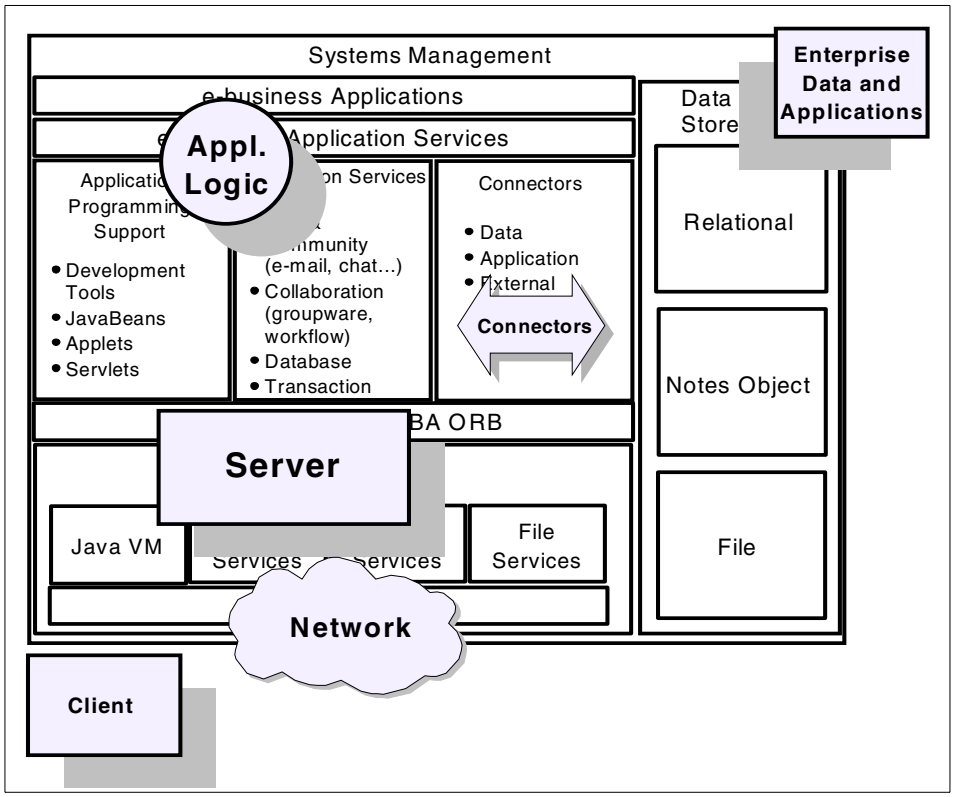


Figure 28. Mapping Building Blocks to IBM's e-business Application Framework

6.3 Competitive Frameworks

Other software companies have used a similar approach to IBM in describing their marketing strategy. Reviewing the frameworks developed by these companies shows many similarities to IBM's e-business Application Framework. Reviewing the competitive frameworks provides an opportunity to see how different vendor products are mapped to Web services and which have capabilities of performing similar functions. A brief introduction to the frameworks offered by Oracle, Netscape and Microsoft can help you position the competitive products you will likely encounter. The competitive information presented here is based on information published on the respective vendor's Web pages. These frameworks are:

- Oracle Network Computing Architecture (NCA)
- Netscape ONE
- Microsoft Windows Distributed interNet Applications Architecture (Windows DNA)

6.3.1 Oracle NCA

NCA is a cross-platform environment for developing and deploying network-centric applications for both the Web and the corporate enterprise. NCA consists of three distinct tiers: the universal data server, the application server, and a universal (any) client.

- Universal data server - Oracle8 provides a product for data management, the universal data server for Network Computing.
- Application server - Web Application Server 3.0 provides a listener-independent application development and deployment environment for the Web.
- Universal client - Any client device can be used to access applications and/or information within NCA. Devices include traditional PCs, Java and/or browser-based clients, mobile devices and Network Computers, such as those from Oracle subsidiary Network Computer Inc. (NCI).

According to Oracle, Network Computing Architecture provides the following key differentiators:

- It is a cross-platform framework based on CORBA/IIOP with support for COM, HTTP and Java.
- It embraces all desktops (Microsoft, Netscape, Network Computer).
- It embraces all operating systems.

- It is independent of Web servers.
- It brings robust features (transactions, messaging, connectivity) to all platforms.
- It is based on accepted open standards.

At the center of NCA is the concept of *cartridges*. Cartridges are software components or business objects which combine small-grained objects into useful components. They comprise applications or components of applications. They can be written in any number of languages including Java, PL/SQL, Perl and C. They take advantage of the CORBA interface and enable the expansion and addition of functionality at any one of the tiers of NCA.

Figure 29 on page 102 shows an example of Oracle's NCA framework.

.

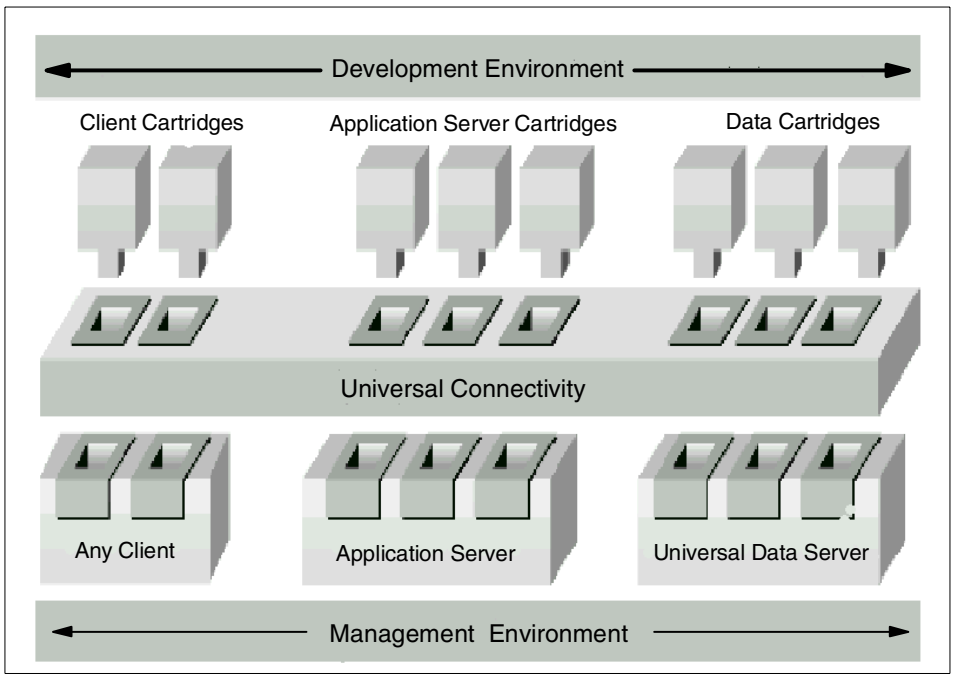


Figure 29. Oracle NCA

6.3.2 Netscape Open Network Environment (ONE)

Netscape ONE is a platform-independent application development and deployment environment. It provides a common application development architecture, integrating a wide range of software and tools, and unifies them

into a single platform. Netscape ONE also provides technologies including JavaScript, Internet Inter-ORB Protocol (IIOP) support, and the Internet Foundation Classes.

Some tools that will incorporate Netscape ONE technologies include:

- Asymetrix SuperCede for Java, which is a visual, interactive development environment.
- SuperCede, which is an interactive and compiled Java development environment that allows developers to modify running applications in real time.
- Borland IntraBuilder, which is an integrated suite of visual tools that combines Borland's tools with JavaScript.
- NetObjects Fusion, which is a Web site authoring application tool that provides a visual drag-and-drop interface to let authors easily include Java applets, JavaScripts, and Netscape Plug-ins in their Web pages.
- Adobe PageMill, which is a WYSIWYG authoring program for creating Web sites.
- Symantec Cafe, which provides a visual development environment allowing Java programmers to create and debug Java applets and applications.

Figure 30 shows an example of Netscape’s ONE framework.

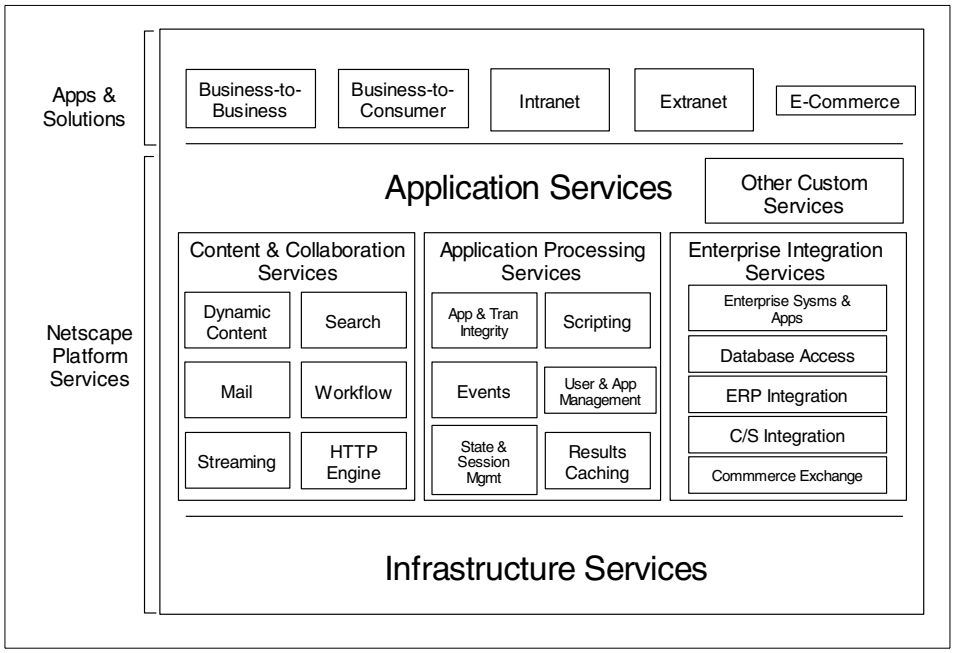


Figure 30. Netscape ONE

6.3.3 Microsoft Windows DNA

Windows DNA is the application development model for the Windows platform. Windows DNA specifies how to: develop distributed applications using the Windows platform; extend existing data and external applications to support the Internet; and support a wide range of client devices. According to Microsoft, because Windows DNA relies on a comprehensive and integrated set of services provided by the Windows platform, developers are free from the burden of building or assembling the required infrastructure for distributed applications and can focus on delivering business solutions.

The Windows DNA architecture consists of a set of system services and component-based application services that support open technology standards, all exposed through the Component Object Model (COM).

- Presentation services (HTML, DHTML, scripting, components, Win32 API)
- Application services (Internet Information Server, MSMQ, MTS, COM+)
- Data services (ADO, OLE DB)
- System services (directory, security, management, networking and communications)

Figure 31 shows an example of Microsoft's DNS framework.

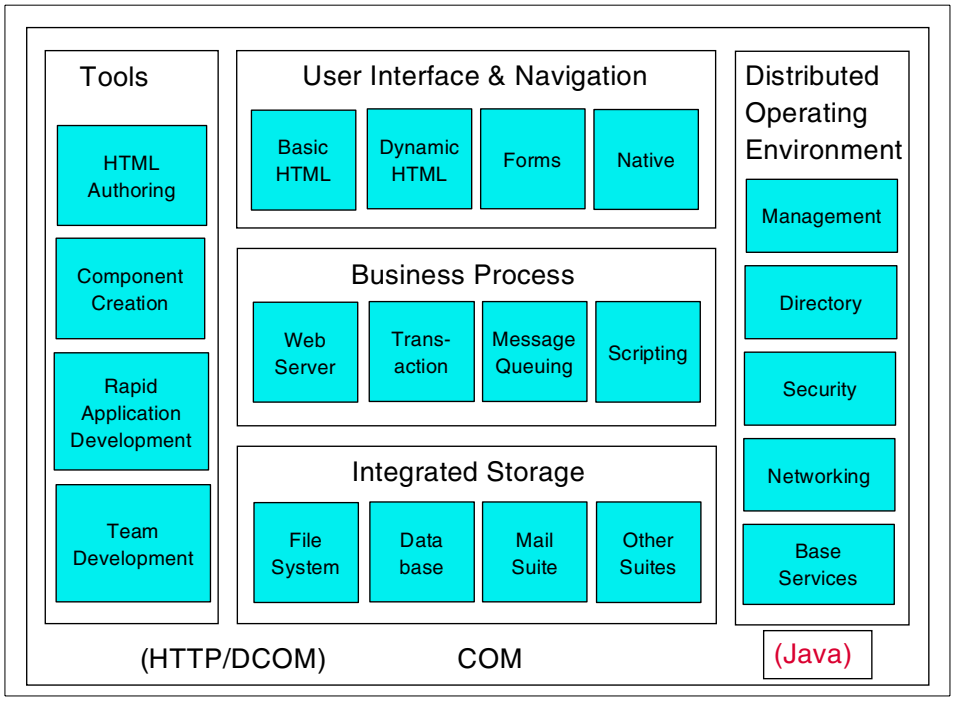


Figure 31. Microsoft DNA

6.4 List of Most Common Components

Table 31 through Table 37. do not provide an all-inclusive list, but rather list those components most often seen in e-business solutions. For each component, the column "Supported Technologies" provides you with guidance on the technologies discussed in Chapter 4, "Develop Architectural Alternatives" on page 39, which the product supports. With each product description, a URL reference is given so you can do further research.

Note

The information in the component tables is a summary, taken for the most part from Web sites. It may be incomplete and is subject to change. For a full and current description of the various components, consult the referenced Web sites.

Using the architecture that you selected in Chapter 5, “Choose Architectural Alternative” on page 81, and the framework defined in 6.2, “IBM e-business Application Framework (ebAF)” on page 96, you can match up products which support the technologies needed in your e-business solution. Based on your experience and expertise in certain product areas, you can use this approach to categorize additional components into the appropriate building block as a basis for discussion with your customer.

Focusing on the Security building block, Table 30 provides component names along with supported technologies and a brief description of the component. Use this information to help determine if you have sufficient information to select a component or, if not, as a reference to where you can obtain additional information.

Table 30. Security

Component Name	Supported Technologies	Description
SecureWay	CDSA SET SSL IPSEC RSA Digital Signatures Firewalls	IBM SecureWay offers a broad portfolio of security software, hardware, services and consulting. You can choose as little or as much security as you need. Good security depends on the right level of investment. It is a trade-off between cost and other issues. Many of the products discussed in this phase are part of IBM’s SecureWay software portfolio. To fully understand how to structure your hardware, software, and services offerings relative to security refer to: http://www.ibm.com/Security

Focusing on the System Management building block, Table 31 provides component names along with supported technologies and a brief description of the component. Use this information to help determine whether you have

sufficient information to select a component or, if not, as a reference to where you can obtain additional information.

Table 31. System Management

Component Name	Supported Technologies	Description
Tivoli	SNMP DMTF-CIM	Tivoli Enterprise provides management solutions that make it easier for large and small organizations worldwide to centrally manage all of their corporate computing resources. For e-business solutions look especially at Tivoli Cross-Site for Security, Tivoli Cross-Site for Availability, and Tivoli Cross-Site for Deployment. Each application includes a newly developed, lightweight framework that manages Internet-based business activities that are outside of traditional boundaries. Reference: http://www.tivoli.com/o_products/html/products_graphics.html
Tools for performance, activity reporting and billing		At this point, note the system management requirements that you defined in Chapter 5. At the server selection stage, check whether the chosen server provides the measurement and reporting tools that you need. If not, you will either have to look for additional products or revise the requirement definition.

Focusing on the Client building block, Table 32 provides component names along with supported technologies and a brief description of the component. Use this information to help determine whether you have sufficient

information to select a component or, if not, as a reference to where you can obtain additional information.

Table 32. Client

Component Name	Supported Technologies	Description
Netscape Navigator	HTML HTTP, HTTPS Applets JavaScript RMI IIOP, XML IMAP4, MIME, S/MIME	Navigator is one of the two leading browsers. Mail and News clients are packaged with the browser, as are media players and the Netcaster component which supports push technology for delivery of information. The Netscape Communicator package adds a Calendar function, 3270 host access using IBM Host-on-Demand and an HTML page editor called Composer. Reference: http://www.netscape.com/browsers
Microsoft Internet Explorer	HTML HTTP, HTTPS Applets JavaScript RMI (must be configured) IIOP, XML IMAP4, MIME, S/MIME	Internet Explorer 4.0 is an integrated suite of Internet software that includes browser, mail and news clients, real-time support through the NetMeeting, NetShow and Chat components, Front Page Express for basic HTML editing and support for push technology using Channels. Microsoft's vision: complete integration of the Internet and the PC. Reference: http://www.microsoft.com/windows/ie/ie40/default.htm
Lotus Notes Client	Collaboration and e-mail HTML, HTTP, HTTPS, POP3, SMTP, IMAP4, MIME, S/MIME plus proprietary protocol over IP, SPX, NetBios or AppleTalk	Lotus Notes is the leading net client software solution that brings all your information together into a single, easy-to-use environment. Notes can be used to communicate with colleagues, collaborate in teams, and coordinate vital business processes. In addition to communicating with a Domino server, you can use the Notes client to communicate with industry standard mail, news and directory servers Reference: http://www.lotus.com (select **Products**, then **Notes**)

Component Name	Supported Technologies	Description
IBM Network Station	HTML HTTP Applets JavaScript	The Network Station is a low-cost computer that provides Internet or intranet access through a Web browser and a Java Virtual Machine for running Java applications. Other names used to describe network computers are thin clients, Java stations, Java machines and net boxes. IBM Network Station offers a friendly graphical user interface and connection to a variety of boot servers such as AS/400, RS/6000, S/390 and Windows NT. It offers native 5250 and 3270 emulators as well as X-Windows terminal sessions. Reference: http://www.pc.ibm.com/networkstation/

Focusing on the Network building block, the Table 33 provides component names along with supported technologies and a brief description of the component. Use this information to help determine if you have sufficient information to select a component or, if not, as a reference to where you can obtain additional information.

Table 33. Network

Component Name	Supported Technologies	Description
IBM Global Services - Network Outsourcing Services		Network Outsourcing Services, rated one of the world's best by Network World, enable you to integrate the newest networking capabilities with those of your legacy data networks and shift the management of this environment to IBM Global Services. This is especially beneficial if your organization has a complex network environment; data, voice and multimedia, or is experiencing rapid business growth. Reference: http://www.ibm.com/globalnetwork/outscsvc.htm

Component Name	Supported Technologies	Description
IBM Global Services - ISP		IBM Global Services provides a managed Internet and intranet service that offers Web site hosting and management, leaving your customer free to focus on developing meaningful content. Services include consulting services for Web site creation and design, help to get a Web site up and running quickly, a cost-effective outsourcing alternative for your Web site operation, a range of enhanced, scalable packages based on industry-leading technologies, and a security-rich, e-Commerce capability to help your customer expand their sales channels worldwide. Reference: http://www.ibm.com/globalnetwork/intnsvc.htm
IBM e-Network Software	TCP/IP SNA	The IBM e-Network Software is a comprehensive set of software products designed for accessing applications and data across the enterprise which can cut costs, simplify network management and further integrate existing applications with the Web. Reference: http://www.software.ibm.com/enetwork/

Focusing on the Server building block, Table 34 provides component names along with supported technologies and a brief description of the component. Use this information to help determine if you have sufficient information to

select a component or, if not, as a reference to where you can obtain additional information.

Table 34. Server

Component Name	Supported Technologies	Description
Domino Web Application Server	Collaboration HTTP, HTTPS LDAP Applets ActiveX MIME SMTP POP3 IMAP4 NNTP JavaScript Proprietary	Domino Web Application Server: - Integrates enterprise data, business processes, workflow management and collaboration into your Web applications giving you all the tools you need to rapidly build convergent Web applications - Integrates core application services you need for your intranet applications, such as directory services, security, messaging, and replication in a single platform, freeing you from having to write the code yourself - Provides a broad spectrum of enterprise integration options, such as extending enterprise data sources and transaction systems to the Internet or your corporate intranet - Supports a wide array of development tools from Lotus and market-leading tool vendors including, NetObjects Fusion, Symantec Café, Borland Jbuilder, and IBM VisualAge for Java, so that you can leverage your existing skills and investments while developing Domino-based applications - Supports Web standard languages and technologies including HTML, JavaScript and Java Reference: http://www.lotus.com/home.nsf/tabs/domino
Apache	HTTP, HTTPS HTML SSL Servlets MIME	The Apache Project is a collaborative software development effort aimed at creating a commercial-grade and freely-available source code implementation of an HTTP (Web) server. The project is jointly managed by a group of volunteers located around the world, using the Internet and the Web to communicate, plan, and develop the server and its related documentation. IBM is a full participant in the Apache HTTP project. As part of the WebSphere Application Server package, IBM will provide commercial, enterprise-level support for the Apache HTTP Server. Reference: http://www.apache.org/

Component Name	Supported Technologies	Description
WebSphere Application Server, Standard Edition	See Appendix D, “The WebSphere Family of Products” on page 155	WebSphere Application Server, Standard Edition forms the foundation of the entire application server family and offers customers a solution to build, deploy and manage e-business Web sites. The Standard edition provides development and runtime environments for Java servlets (JSDK). Extends JSDK to provide personalization, session tracking, and real-time Web site management. An industry-standard ORB based on CORBA 2.0 is also provided. While Websphere will distribute the Apache Web Server as part of the product, WebSphere will plug into any of the Web servers mentioned in this document. References: http://w3.software.ibm.com/websphere http://www.software.ibm.com/websphere
WebSphere Application Server, Advanced Edition	See Appendix D, “The WebSphere Family of Products” on page 155	WebSphere Application Server, Advanced Edition builds on the Standard edition to provide enhanced support for scaling Web sites into secure, transactional e-business applications. The Advanced edition connects Web applications to existing databases and host-based transaction systems, and offers sophisticated tools to simplify distributed component-based application development. References: http://w3.software.ibm.com/websphere http://www.software.ibm.com/websphere
WebSphere Application Server, Enterprise Edition	See Appendix D, “The WebSphere Family of Products” on page 155	WebSphere Application Server, Enterprise Edition enhances the Advanced edition combining TXSeries, IBM's world-class transactional application environment, with the full distributed object and business process integration capabilities of Component Broker. All applications currently running on TXSeries or Component Broker will be fully supported with the Enterprise edition. References: http://w3.software.ibm.com/websphere http://www.software.ibm.com/websphere

Component Name	Supported Technologies	Description
WebSphere Performance Pack		WebSphere Performance Pack is composed of three major components: an enterprise file system, a caching proxy server and a load balancing/server monitoring tool. These features provide a flexible combination of caching, proxy and filtering functions, file content management and replication, and load balancing into a single Internet hosting infrastructure. Reference: http://www.software.ibm.com/webservers/perfpack
Component Broker	Java CORBA IIOP	Component Broker is an enterprise solution of distributed transactional object middleware, providing a scalable, manageable runtime and visual tools for component-based development combining existing data and business logic with new application capability. Reference: http://www.software.ibm.com/ad/cb
San Francisco	Java RMI	IBM and a collection of leading international software companies are collaborating to build SanFrancisco, a series of business process components and frameworks for building line-of-business server applications in Java. SanFrancisco enables developers to more quickly create industry or function-specific applications for their customers. Reference: http://www.ibm.com/Java/Sanfrancisco/

Component Name	Supported Technologies	Description
Net.Com-merce		Net.Commerce is the IBM commerce server and includes Net.Commerce START and Net.Commerce PRO. Net.Commerce Start allows small and medium-sized businesses to extend their marketing, merchandising, sales and customer support processes to their trading partners and to the Web. Net.Commerce PRO allows large businesses with sophisticated Web sites to link to their supply chain, inventory management and other legacy systems and easily handle large numbers of transactions. Net.Commerce is used for business-to-business applications by customers working with their supply chains, in manufacturing and other business applications as well as for business-to-consumer applications by customers selling to consumers directly over the Web. Reference: http://www.software.ibm.com/commerce/net.commerce/
Netscape Enterprise and FastTrack Server	HTTP, HTTPS SSL LDAP IIOP Servlets MIME	Netscape FastTrack Server is an entry-level Web server that provides an engine for publishing and serving static and dynamic Web pages and offers an environment for building or deploying many types of web or database applications. For content management, centralized administration capabilities, or advanced crossware application services, customers can upgrade to Netscape Enterprise Server. Reference: http://www.netscape.com/servers
Microsoft IIS, Site Server and Site Server (Commerce Edition)	HTTP, HTTPS SSL Proprietary ASP (Active Server Pages) Servlets MIME	Microsoft's Web server, the Internet Information Server (IIS), is available without charge to run on Windows NT Server. At the time of writing, the latest version, which includes the Microsoft Index Server, can be found along with Microsoft Transaction Server, Message Queue Server and Internet Connection services for Microsoft RAS as part of the Windows NT 4.0 Option Pack at: http://www.microsoft.com/windows/downloads/contents/Updates/NT4OptPk/ Enhanced commercially packaged server offerings are Microsoft Site Server (for intranets) and Site Server (Commerce Edition) for Internet commerce applications. For more details see: http://www.microsoft.com/products/os.htm

Component Name	Supported Technologies	Description
INPRISE VisiBroker (formerly Visigenics)	CORBA IIOP Java	VisiBroker provides a CORBA 2.0 ORB environment for building, deploying, and managing distributed Java applications that interoperate across multiple platforms. With native implementations of the Internet Inter-ORB Protocol (IIOP), VisiBroker provides support for the industry-standard protocols for the Internet, intranets, and Web-based computing. Reference: http://www.inprise.com/visibroker
Iona's Orbix	CORBA IIOP Java	Orbix is based on the leading standard for distributed computing; CORBA, the Common Object Request Broker Architecture from the independent group the Object Management Group (OMG). Orbix provides interoperability with Microsoft D/COM through OrbixCOMet. OrbixOTM adds transactions, security and scalability to the CORBA environment. Orbix is available on all major platforms; Windows NT, UNIX, and mainframe MVS. Reference: http://www.orbix.com/products/orbix

Focusing on the Application Logic building block, the Table 35 provides component names along with supported technologies and a brief description of the component. Note that while this is the application logic building block, the focus here is on application development components. Use this information to help determine if you have sufficient information to select a component or, if not, as a reference to where you can obtain additional information.

Table 35. Application and Web Site Development

Component Name	Supported Technologies	Description
WebSphere Studio	See Appendix E, "IBM Application Development Tools Classification" on page 161	WebSphere Studio provides a complete set of tools that are integrated and designed to support all the people who develop Web sites. The content authors, graphic artists, and webmasters can all work on the same projects, each having access to the files they need. Reference: http://www.software.ibm.com/webservers/studio

Component Name	Supported Technologies	Description
Visual Age for Java	Java JDBC ActiveX (as add-on) Servlets JavaBeans EJB	VisualAge for Java, an Integrated Development Environment (IDE) that seamlessly integrates code browsing, editing, execution, and debugging. The IDE includes wizards and visual builders for creating and visually editing code. VisualAge for Java provides support for the development of servlets, JSPs, and EJBs. Reference: http://www.software.ibm.com/ad/vajava
Enterprise Java Beans (EJB)	Java	EJB is a component architecture for multi-tier, distributed Java applications, which extends beyond normal JavaBeans to support server components. The EJB infrastructure provides transactional and system services for the application components, making distributed, client/server applications easier to develop, deploy, manage, and maintain. The multi-tier approach increases the performance, scalability, and reliability of an application. The beans which are built to this specification are called Enterprise JavaBeans. The distributed approach makes them location-independent, so a system administrator can move them around to reconfigure system load. IBM and other suppliers will be delivering specific EJBs because of the flexibility they provide in the construction of middleware. To find more information on Java, JavaBeans and EJB. Reference: http://www7.software.ibm.com/vad.nsf/
Notes Designer for Domino	Java HTML CORBA IIOP Proprietary Components	Notes Designer for Domino provides an environment to quickly develop secure, collaborative, business applications for the Net. Notes Designer for Domino offers developers and Web designers: ability to access features like views and image maps; ability to write Domino server agents in Java embedding Java applets directly into Notes forms and documents, and alter their properties quickly and easily. Reference: http://www.lotus.com (select **Products**, then **Notes Designer for Domino**)

Component Name	Supported Technologies	Description
NetObjects Bean Builder (formerly Lotus Bean Machine)	Java JavaBeans applets	BeanBuilder lets you create Java applets and beans quickly and visually, without Java programming. Applets and beans can be visually assembled on screen by using a provided set of JavaBean components or importing any standard JavaBean. Logic can be added allowing the components to interact in the form of “connections,” or simple rules that describe how an event associated with one bean can trigger an action in another bean. Reference: http://www.netobjects.com/products/html/nbb1.html
NetObjects Fusion	HTML applets	NetObjects Fusion provides a development environment with a site-wide approach and automates most Web site tasks. Fusion provides expanded page layout edition options, a visual message-based environment for creating sophisticated DHTML animations, full directory control when publishing to the Web site, and cross-browser compatible HTML. Reference: http://www.netobjects.com/products/html/nof.html In addition, NetObjects TeamFusion supports working in teams to develop a Web site and the Domino Fusion Connector allows a site built with Fusion to be published to a Domino server.
Lotus FastSite	HTML, Java	FastSite, packaged with Lotus SmartSuite Millennium edition, provides a quick and easy way to build a Web site from files created with any SmartSuite component or Microsoft Word 97 or PowerPoint. This site can be stored on a Web server or published to Domino. Reference: http://www.lotus.com/fastsite
Symantec Visual Cafe Pro	Java JDBC (as add-on) JavaBeans Servlets	Visual Café Professional Edition includes a library of professionally developed JavaBeans with source code and Wizards for Servlet development, localization and deployment. The Java Code Helper assist with Java editing and code integrity. Reference: http://www.symantec.com/domain/cafe/vcafe30.html

Component Name	Supported Technologies	Description
INPRISE JBuilder (formerly Borland)	Java JDBC JavaBeans Servlets CORBA IIOP	Included in JBuilder: JDK Switching, CodeInsight, BeansExpress, BeanInsight, Pure Java DataExpress components, JDBC connectivity, grid control with source, 200+ JBCL and JFC/Swing beans with source, data-aware dbSwing beans with source, Servlet Wizard, command-line tools, and Local InterBase Server. Reference: http://www.inprise.com/jbuilder/
Sybase PowerJ	Java JDBC ActiveX	PowerJ is a development environment for Java programmers. It combines productive Java with graphical HTML features, for enterprise Web applications. PowerJ also includes AppModeler for application and database design and a development edition of the Enterprise Application Server for deploying distributed applications. Reference: http://www.sybase.com/products/powerj
Microsoft Visual J++	Java ActiveX	With Visual J++ 6.0, developers can construct Windows-based applications and components using two-way RAD tools and the Windows Foundation Classes (WFC), an object-oriented framework that encapsulates and simplifies the Win32 API. Reference: http://www.microsoft.com/products

Focusing on the Connectors building block, Table 36 provides component names along with supported technologies and a brief description of the component. Use this information to help determine if you have sufficient

information to select a component or, if not, as a reference to where you can obtain additional information.

Table 36. Connectors

Component Name	Supported Technologies	Description
CICS Gateway for Java	Java Applet Java Application Servlet Proprietary RPC	The IBM CICS Gateway for Java provides a way to access the established transaction capabilities of CICS servers from any Java program. A Java applet or Java application in the Web client can directly call CICS programs and data simply by invoking the small Java class supplied with the Gateway. The Gateway is a multi-thread Java application, which interfaces with a co-resident CICS Client to communicate with CICS servers. The Gateway can handle many asynchronous requests from separate Web clients, and by using the capabilities of the CICS client can communicate with many CICS servers concurrently. Reference: http://www.software.ibm.com/webservers/connectors/
CICS Internet Gateway	HTML HTTP Proprietary RPC	The IBM CICS Internet Gateway provides an interface between a Web server and a CICS application, allowing conversion of 3270 data streams into HTML format used by the World Wide Web. The CICS application can then be accessed by any Web browser with no change to either the browser or the CICS application. Security functions provided by the browser and by the CICS host are passed through the gateway. Reference: http://www.software.ibm.com/webservers/connectors/
MQSeries Client for Java	Messaging Java applet Servlet	MQSeries Client for Java enables Java applets and Java applications to issue calls and queries to MQSeries, allowing users to become true participants in transactions. It provides access to mainframe and legacy applications over the Internet without any additional MQSeries code on the machine. With MQSeries Bind for Java, Java applications running on the same server as MQ Manager can access MQ resources by binding directly with the MQ Manager without incurring the network overhead associated with MQSeries Client or Java. Reference: http://www.software.ibm.com/webservers/connectors/

Component Name	Supported Technologies	Description
MQSeries Internet Gateway	Messaging HTTP	MQSeries Internet Gateway takes a request originating from a Web browser and converts into an MQSeries message. The message will be delivered by MQSeries to a serving application running on any one of over 20 platforms that are supported by MQSeries. The browser can wait for a response to be returned from the application via MQSeries and the gateway, or alternatively an acknowledgment can be sent to the browser by the gateway as soon as it has received the request, confirming to the user that work is in progress and freeing that browser session to perform other tasks. Reference: http://www.software.ibm.com/webservers/connectors/
eNetwork Host On-Demand	Applet Telnet 3270	eNetwork Host On-Demand is a connectivity solution that provides standard Telnet application discovery and access through the WWW, and is a cost-effective alternative to desktop 3270 terminal emulation. It provides a low-cost solution for Intranet and Web users who need access to their central computer applications or databases from any Java-enabled end-user platform solution. Reference: http://www.software.ibm.com/webservers/connectors/
IMS Web	Proprietary	IMS Web uses TCP/IP communication infrastructure and middle-tier Web servers to make IMS transactions available to the Web. With IMS Web, developers can dynamically build transaction-specific connectors to the Web for a variety of environments. These connectors can provide access to IMS applications and data in the S/390 environment. Reference: http://www.software.ibm.com/webservers/connectors/ (select **IMS Connectors**)
IMS WWW Templates	Proprietary	IMS WWW Templates contain templates for providing Web access to IMS programs that accept APPC connections. The templates are invoked from the Web server on any APPC platform (such as OS/2, Windows, AIX, Sun Solaris, or OS/400) and provide the conversion between URL and HTTP information and APPC protocols. Reference: http://www.software.ibm.com/webservers/connectors/ (select **IMS Connectors**)

Component Name	Supported Technologies	Description
IMS Client for Java	Proprietary	IMS Client for Java provides sample code for preparing a Java program to access IMS applications and data running in a S/390 environment. It also includes a user exit to translate the message into the format required to access IMS through the IMS TCP/IP OTMA connection. It can be used from any Java Virtual Machine environment. The Java application accesses IMS applications and provides the information back to the Java Virtual Machine. Reference: http://www.software.ibm.com/webservers/connectors/ (select **IMS Connectors**)
IMS TCP/IP OTMA Connection	Proprietary	The IMS TCP/IP OTMA Connection provides enhanced communication linkages between remote workstations and IMS. It provides improved performance and access for TCP/IP users needing access to IMS applications and data. It supports multiple TCP/IP clients accessing multiple IMS systems. The IMS TCP/IP OTMA Connection provides access from any TCP/IP-supported environment to IMS on a S/390 platform. Reference: http://www.software.ibm.com/webservers/connectors/ (select **IMS Connectors**)
DCE Encina Lightweight Client (DE-Light)	RPC HTTP	The DE-Light product family extends the operability of a company's DCE and Encina-based installations to Web access. DE-Light provides the ability to leverage the Internet using Kerberos and SSL security. DE-Light is implemented using an intermediate gateway installed on a DCE-enabled machine to process client requests. The client and gateway communicate using TCP/IP or HTTP. Reference: http://www.transarc.com/Product/Txseries/DELight/index.html

Component Name	Supported Technologies	Description
Domino NotesPump	Proprietary	Domino NotesPump data distribution system provides secure, scheduled data distribution and replication services between all supported data sources. NotesPump provides a highly scalable architecture which is multi-tasking and SNMP manageable. Web and Lotus Notes clients can interactively access or update host and RDBMS data sources using the new NotesPump Realtime Activity support. Note: NotesPump is being renamed to Lotus Enterprise Integrator (LEI). Reference site for Domino Enterprise Connectivity: http://www.eicentral.lotus.com/
DECS	Proprietary	Domino Enterprise Connection Services (DECS), a standard feature provided with Domino 4.6.3 and 5.0, provides forms-driven (no programming) connectivity between Notes applications and several leading back-end relational databases including DB2/UDB, Oracle and Sybase. Reference: http://www.eicentral.lotus.com/eibu_knowbase.nsf/eipages/knowmap
Domino LSX for SAP R/3	Proprietary	Domino LotusScript Extension (LSX) for SAP's R/3 allows access to ERP data by Domino web applications. Direct access to SAP's R/3 data via a LotusScript Extension (LSX) providing Notes developer access to SAP R3 APIs including; the Remote Function Call libraries and SAP's R/3 BAPIs, linking to 170 Business Objects. Domino provides a customizable interface for this powerful ERP system. This LSX for R/3 allows for cross-platform, bi-directional data exchange and is capable of both synchronous and asynchronous data transfer activity. Low-level integration access via the RFCs and BAPIs provide BCD Batch Input functions for data input and reporting functions for data export. High-level integration access is provided through the SAP Automation Server Classes for OLE2. Reference: http://www.edge.lotus.com/eibu_knowbase.nsf/eipages/r3home

Component Name	Supported Technologies	Description
Domino MQ LSX	Messaging	Domino MQSeries Link LotusScript Extension (MQLSX) gives your Lotus Notes and Domino applications, written in LotusScript, the ability to communicate with applications running in non-Notes environments, using MQSeries. The MQLSX is an Application Programming Interface that you call from LotusScript to access the MQSeries Message Queue Interface (MQI). Reference: http://www.edge.lotus.com/eibu_knowbase.nsf/eipages/mqcicshome
Domino CICS and MQSeries Connectors	Messaging RPC	Domino CICS and MQSeries Connectors allow access to more than 25 different platforms such as HP-UX, Tandem and Digital via MQSeries from IBM; and subsequently communicate with the transaction systems they support, including CICS, IMS, MVSÍ, and so on. You can also access CICS directly. MQEI, a component of CICS and MQSeries Connectors, provides the facilities to integrate these applications into your Notes or Domino environment, by giving developers a single set of LotusScript classes that provide a simple, common API for enterprise access. Reference: http://www.edge.lotus.com/eibu_knowbase.nsf/eipages/mqcicshome
Domino DB2 LSX	Proprietary	DB2 LotusScript Extension (DB2 LSX plug-in) provides native access to DB2 through the DB2 CLI Interface. Based on the LotusScript Data Object, the DB2 LSX is a set of three classes that access DB2 data as well as expose DB2 extended functionality, such as support for DB2 user-defined types and DB2 large objects. Reference: http://www.eicentral.lotus.com/

Component Name	Supported Technologies	Description
Net.Data	Proprietary Java HTML	Net.Data is listed here because it is considered by IBM as a connector, but it could also be included in Application Development since you can build an application using the capabilities of Net.Data. Net.Data is an application that allows Web developers to easily build dynamic Internet applications using Web Macros. Net.Data Web Macros have the simplicity of HTML with the power of dynamic SQL. Net.Data provides database connectivity to a variety of data sources, including information stored in relational databases and flat files. Your data sources, such as DB2, Oracle, and Sybase, can be on a wide range of operating systems. Reference: http://www.software.ibm.com/webservers/connectors/

Focusing on the Enterprise Data and Applications building block, Table 37 provides component names along with supported technologies and a brief description of the component. Use this information to help determine if you have sufficient information to select a component or, if not, as a reference to where you can obtain additional information.

Table 37. Enterprise Data and Applications

Component Name	Supported Technologies	Description
DB2	Proprietary	DB2 offers open, industrial-strength database management for business intelligence, transaction processing, and a broad range of applications for all types of businesses. Reference: http://www.software.ibm.com/data/db2/
Transaction Server	HTTP Java Proprietary	Transaction Server (CICS) providing business transaction services (BTS), combined with technologies like CORBA and Java, allow improved productivity for application developers and reduced production costs resulting in the capabilities for attracting new customers via the Web. Reference: http://www.software.ibm.com/ts

6.5 Information for Component Selection

At this point, you should have a collection of information available concerning the e-business solution you are designing including:

- Business and technical requirements
- Information about the customer's current environment
- Resulting documentation from architectural alternatives and architecture selection
- A selected architecture
- Brief description of possible components to support your architecture
- Some of the technologies that the components support

As pointed out in 6.2, "IBM e-business Application Framework (ebAF)" on page 96, building a customer e-business framework is one technique that can help lay out your customer's e-business capabilities. Based on the selected architecture for the e-business application you are designing, you can determine the areas where the customer does not have capabilities to support the new application or areas where existing products may not be sufficiently robust to support the new application.

As you complete your e-business solution, now is the time to add products to the solution. In some cases, you may have no choice of selection; for example, the customer may have dictated that NT be the hardware platform. However, following the approach offered in this redbook, you may have earned considerable credibility regarding e-business solutions. This credibility will earn you the right to propose IBM content in support of your solution. IBM content can be categorized into the following:

- Hardware
- Software
- Services

6.5.1 Hardware Platform

Up to this point, the e-business solution should have been completed without regard to any particular hardware platform. However, certain business and technical requirements should be re-evaluated to determine if a particular platform provides any specific advantage. Business and technical requirements that could effect this evaluation include:

- The solution must be highly available.

- The solution must be scalable.
- The solution must meet certain performance expectations.
- The need to leverage existing skills.
- The manageability of the environment.
- The security requirements.
- The number of concurrent users projected for the application.
- The integration into current enterprise data and applications.

Evaluating these requirements against IBM's platforms is not an exact science. In most cases, specific customer requirements such as existing skills will point to a specific platform. In other cases, the customer may already have made platform choices for their enterprise and you will need to adhere to that previous decision. However, if these decision points do not impact the selection, then requirements such as number of concurrent users, performance expectations, security requirements, and integration into current enterprise data and applications will help point to a specific platform. Today's platform choices are IBM's Netfinity Servers, RS/6000 Servers, AS/400 Servers, or S/390 Servers.

If you have an opportunity to market IBM hardware to support your e-business solution, you should look for more information concerning IBM's hardware offerings (see Appendix H.5, "URLs" on page 176). At this point, engaging a marketing specialist for the target platform might be appropriate.

6.5.2 Software

Using the product descriptions in 6.4, "List of Most Common Components" on page 105 and the technologies defined in the selected architecture, you can start selecting software products to support your e-business solution.

By packaging the more commonly used products into the three WebSphere products (see Appendix D, "The WebSphere Family of Products" on page 155), IBM has simplified the product selection process. These products are:

- WebSphere Application Server, Standard Edition
- WebSphere Application Server, Advanced Edition
- WebSphere Application Server, Enterprise Edition

However, if the customer has a specific requirement for using another vendor's product within the solution, you will need to know the capabilities of the individual components that make up these WebSphere products.

Appendix D, “The WebSphere Family of Products” on page 155 provides details on the packages of the WebSphere product suite.

Another aspect of software selection that you will want to include is the selection of a development environment. Most likely, discussion of the development environment will have surfaced in the technology discussions surrounding the architectural alternatives. Certainly, the development environment might be critical if your solution requires deployment across multiple platforms. For example, can you develop on one platform and deploy on any heterogeneous platform?

As with WebSphere product packaging, IBM has simplified the selection of a development environment by combining many aspects of developing applications for the Web into one product, WebSphere Studio. VisualAge for Java is being distributed with WebSphere Studio to provide a complete programming environment for Web Applications. Appendix E, “IBM Application Development Tools Classification” on page 161 contains more information on the products that make up WebSphere Studio and what part of Web development these products address.

6.5.3 Services

Throughout this redbook, references have been made to the possibility of additional work that needs to be accomplished to help understand a customer’s environment or to update his environment. Examples of this work include:

- Detailed investigation of enterprise data and application to better understand interfaces, data structures, application logic, and how much additional coding might be necessary to support an e-business solution.
- Detailed network analysis to gather baseline data, provide design assistance, and problem resolution assistance in order to insure the network can support an e-business application.
- Detailed security analysis to better understand security requirements and make recommendations on how to incorporate the right level of security into your e-business solution.
- Additional education and training so that customer personnel can be ready for an e-business solution.
- Outsourcing the operation and management of the Web site and network. These are commonly referred to as Internet Service Provider (ISP) and Value Added Network (VAN) services.

- Engaging a services organization to manage and implement the proposed e-business solution.

If you have an opportunity to market IBM services to support your e-business solution, you should look for more information concerning IBM's services offerings (see Appendix H.5, "URLs" on page 176).

6.6 Gaining Knowledge of Components

To be successful in designing e-business solutions, you need to be versed in the technologies and products that support typical e-business solutions. While this redbook provides some guidance, it is not meant to provide you the in-depth technology and product knowledge needed to design e-business solutions. You should take advantage of some or all of the following activities to assist you in gaining the needed expertise:

- Hands-on experience is one way to gain knowledge of products and technologies. Many of these products are available over the Internet for evaluation.
- Training by attending appropriate available courses (see Appendix H.7, "Education and Training" on page 178).
- Reading about technologies and products to keep your knowledge base updated (see Appendix H.4, "Other Resources" on page 175 for additional books and URLs).
- Being mentored by a more experienced I/T Specialist or I/T Architect.
- Participation in an ITSO Residency.

6.7 Determining if POC or TIP is Needed

At this point, you should have an e-business solution. Now you should consider other activities that might be needed to help clear up any further concerns or uncertainty about your solution. You might consider the following two activities:

- Proof of Concept (POC): details can be found in Appendix A, "Proof of Concept (POC)" on page 133
- Testing for Indicators of Performance (TIP): details can be found in Appendix B, "Test for Indicators of Performance (TIP)" on page 139

6.7.1 When to Conduct a POC

Does a POC involve hard work and expense? You bet. Do you conduct a POC for every opportunity? Certainly not. So under what circumstances would you consider planning and performing a formal Proof of Concept?

- If the proposed solution is in large part new and untested technology
- If the proposed solution is a mix of new and existing technology, and the interaction between the two has not been proven
- If the choice between architectural alternatives is not clear, and some method of gaining consensus on a choice is necessary
- If, regardless of the stability of the proposed technology, the customer displays discomfort with moving forward with the solution

It is not possible to provide a concise formula for arriving at a yes or no answer as to whether to conduct a POC. Many factors come into play, including the issue of who will bear the expense of the POC. In the end, it is a judgment call that must be made by the solution architect.

6.7.2 When to Conduct a TIP

The considerations for conducting a POC can be applied to a decision to conduct a TIP. Additional considerations for conducting a TIP would be if preliminary performance estimates suggest that more detailed work was necessary or there was concern that:

- Some portion of the design might be too costly in terms of allocated end-to-end budget.
- The selected technologies and products might not scale with this design.
- The selected platform might not support the number of projected concurrent users.

As with the decision to conduct a POC, conducting a TIP is not a simple process and can be costly. However, detecting deficiencies in the design, technologies chosen or components selected can save time and money in the long run.

6.8 Chapter Checkpoint

At this point you should have selected products that will support your architecture and are ready to make a formal proposal to your customer. Before proceeding, ask yourself the following questions:

1. Have you gained commitment from your customer on the products selected?
2. Have you documented your design?
3. Have you considered all the options for providing services?
4. When selecting products, did additional questions surface concerning technologies included in your design?
5. Have you reviewed your design with appropriate specialists or architects?
6. Are all of the products selected generally available?

If you can answer yes to each question, congratulations! You should have an e-business solution agreed to by your customer from which you can make a formal proposal. Otherwise, you may need to go back and repeat some of the steps in earlier phases.

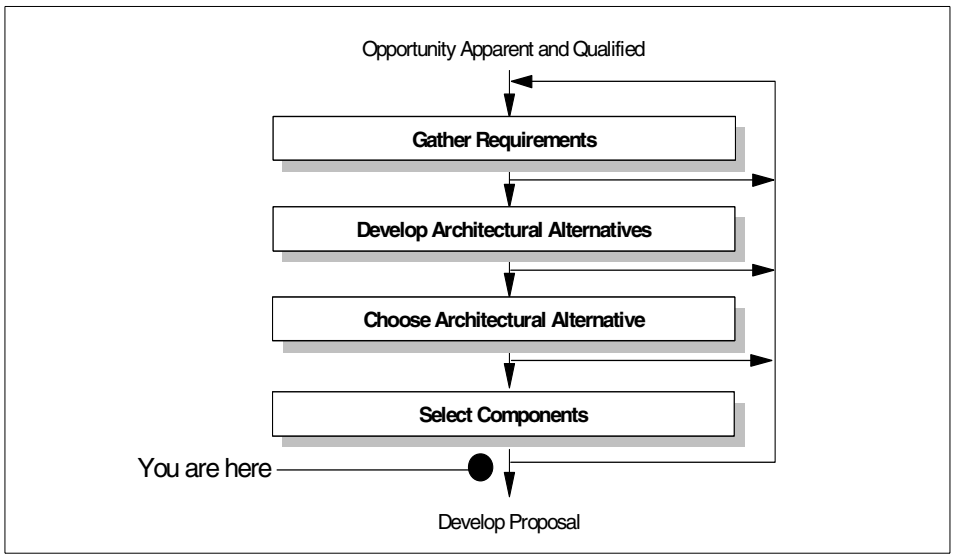


Figure 32. This is Where You Are in the Process

Chapter 7. Summary

Throughout this redbook, we have attempted to strike a balance between our desire to present e-business designs as relatively simple, and the reality of how complex they can sometimes become. To do this, we used a simple diagram to represent e-business, and we called this diagram the building blocks of e-business (see Figure 33).

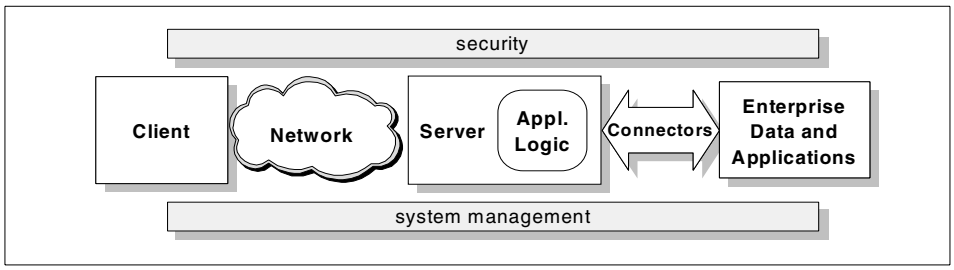


Figure 33. Building Blocks of e-business

However, we made reference to more complex architectures and frameworks, such as in 4.5.2, “IBM’s Intellectual Capital and Our Building Blocks” on page 74, where we pointed you to existing designs in use by IBM’s consulting and services people, and in 6.2, “IBM e-business Application Framework (ebAF)” on page 96, where we introduced the IBM e-business framework.

The technology involved with e-business is rapidly changing. That change brings about greater opportunity for solutions, as well as added complexity. New technologies enter the picture before their relationships with existing technologies have been fully explored or tested. However, when you reduce it all to its most basic essence, you come back to our building blocks.

You have customers seeking e-business solutions, and IBM has the building blocks for those solutions. This redbook presents the recipe for success, which comes in the form of five recommendations:

1. Use the approach presented in this redbook to gather requirements and formulate a solution. It is not intended to be a step-by-step checklist; rather, it helps you blend your experience with the information presented to come up with a solution.
2. Key off the requirements you gather from the customer, for that is what will drive their acceptance of your solution. Remember, customer requirements should drive the e-business solution. Do not let products or technology drive the solution.

3. Surround yourself with people skilled in the functional areas of e-business. Few people know everything about everything, so building a team with experts in the various areas will go a long way to ensuring success.
4. Reuse proven designs where you can, because building a solution from scratch is a difficult and expensive undertaking. IBM is collecting knowledge of past solutions that were successful, and to the extent you can reuse one of those, it will help drive your success.
5. When faced with very large or very complex solution designs, call in someone familiar with one of IBM's methodologies. Complex environments require a careful discipline in understanding the issues and proposing a solution. Appendix F, "Other Design Methods and Sources of Information" on page 163 lists these other methods and what they provide.

IBM has a vast array of components and technologies to apply to an e-business design, perhaps the best in the industry. The solution design for your customer is out there; go find it.

Appendix A. Proof of Concept (POC)

The objective of a Proof of Concept (POC) is to validate, in whole or in part, a proposed solution. By conducting a POC, the solution's logical and physical architecture, connectivity, data access, platforms and tools all work together as a system. More importantly, a successful POC instills confidence in the parties--the IBM team and the customer--to move forward with the solution.

A Proof of Concept does *not* attempt to test and validate every aspect of the solution design. It is tempting, when planning a POC, to include as much as possible. Avoid that temptation. This is discussed in more detail in A.1.1, "What Will Be Tested, How It Will Be Tested" on page 133.

A.1 Preparation

Before beginning a POC, you should spend time developing answers to the following questions:

- What will be tested and how will it be tested?
- What will be the degree of customer involvement in the design, construction and execution of the POC?
- Have you gained the customer's agreement on key aspects of the POC?
  - Has the customer agreed with you that one should be conducted?
  - Has the customer agreed with you on what should be tested? This would include the functional components to be tested and the test plan with agreed to use cases.
  - Has the customer agreed with you on what will constitute success or failure of the POC?
  - Has the customer agreed with you on what they will do if the POC is a success or a failure?
- Who will bear the expense of the POC?

A.1.1 What Will Be Tested, How It Will Be Tested

A Proof of Concept should not try to test every aspect of the proposed solution. Testing to that level of detail is reserved for the implementation phase of a project.

Instead, a POC should focus on validating the high-level functional aspects of a design. For example, you might consider a proposed solution that involves browser access to a Web server that will use MQSeries to communicate with a CICS subsystem on an OS/390 host system, as shown in Figure 34.

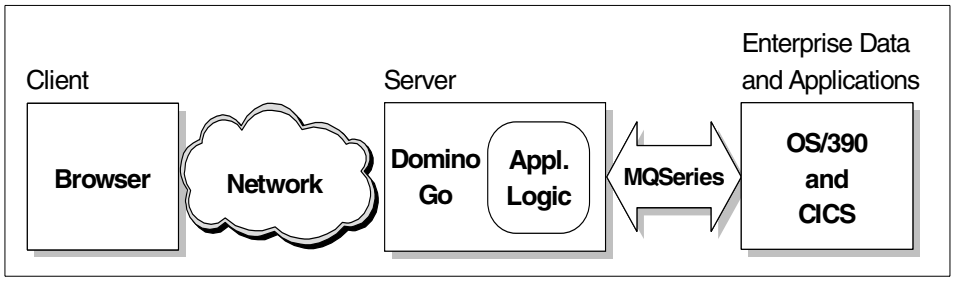


Figure 34. Example Solution Design Under Consideration for POC

Keeping in mind that the objective of a POC is to validate the high-level functionality of the components and their interfaces, you might consider one possible approach to defining what to test and how to test it, as shown in Figure 35.

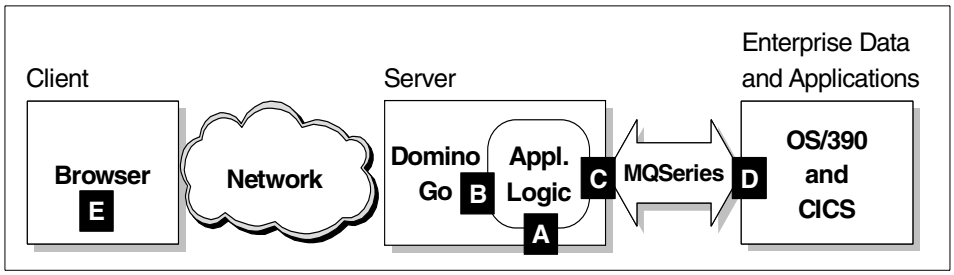


Figure 35. Example POC Test Design

The construction and validation might follow a course something like this:

A Prototype of application logic

This prototype does not need to be functionally complete. Rather, it should provide just enough code to interface successfully with the Domino Go Webserver and MQSeries, and perform the high-level logic functions as defined in the solution design.

B Interface between the Domino Go Webserver and application logic

This interface (along with that represented by C and D) represents points where validation of functionality is important. Therefore, the POC should be designed to exercise the major function of this interface.

This does not imply that the Domino Go Webserver would require functional testing. In this example, simply testing the interfaces (B, C and D) drives the Domino Go Webserver sufficiently to validate its functional role in this solution. This is an example of limiting the scope of the POC.

C Interface between the application logic and MQSeries

Your prototype application will call upon MQSeries to communicate with the CICS subsystem on the OS/390 host. You should validate that the functions tested in B of this example invoke MQSeries as expected, and coupled with D, return the results as expected.

Again, to the extent you drive your application logic to request and retrieve information from CICS, the MQSeries product itself will be validated sufficiently for the POC illustrated in this example.

D Interface between MQSeries and CICS subsystem

This interface should act upon *a real* CICS system, and not a mock-up of one. It would be sufficient to drive a test CICS system, but it would not be appropriate to mimic or otherwise simulate CICS through some code written for that purpose.

CICS in this example would not be the focus of validation. Rather, the focus would be the ability of MQSeries to pass the request off to CICS, and then respond back to the application code on the server.

E Browser

All browsers are not made equal. Particularly if your application employs Javascript, you should validate the major players in the browser market.

By validating the browser, you validate the network and the network connection to the server; therefore, further testing of the network would not be called for in this example.

Keep in mind that this is just an example, and that your POC might call for more testing (or less testing) of these components. The point of this illustration is to show the importance of identifying the aspects of the design that, when tested, provide sufficient validation of the overall design to warrant continuing with the project.

A.1.2 Degree of Customer Involvement in POC

This is really a judgment call between the IBM team and the customer. We have found that involving the customer in the design of the POC is helpful in keeping them involved in the planning of the overall solution.

Note that the actual construction of the POC test environment is sometimes best done away from the customer, so that problems can be ironed out without using too much of the customer's valuable time. When the POC test environment is ready, however, the customer should be brought back in and instructed on the process used to construct the test. Finally, the running of the test is best done with the customer present.

Whatever decision is arrived at regarding the degree of involvement, this should be discussed and agreed upon by both the IBM team and the customer.

A.1.3 Customer Agreement

It is not advisable to assume anything when preparing to devote the time and expense to a POC. Therefore, it is important that you gain your customer's agreement on the following topics before you commit to, or start, a POC:

- Whether a POC is desired or warranted

  Any mismatch between your expectations and your customer's expectations regarding the need or lack of need for a POC may cause significant problems later.

- What is to be tested

  In A.1.1, "What Will Be Tested, How It Will Be Tested" on page 133, we cover how you might determine the scope of the POC. Does your customer agree with this plan? If not, then go back and refine it until you gain their agreement.

- What constitutes success or failure

  Determine before the POC how you and the customer will "grade" the results of the POC, and what the next step will be in the project based on the "grade" received. This is a very difficult process, since objective criteria are difficult to define when the exact behavior of the test cannot be predicted.

- What the customer will do after the POC is complete

  If the POC is deemed a success, do you have your customer's agreement that they will continue with the next phase of the project? Similarly, if the POC meets the "failure" criteria, do you have a clear understanding of what the next course of action will be? This seems an obvious question, but it is often overlooked.

If there is one key point here it is this: *do not assume what you have not discussed with the customer*. If at all possible, capture the agreement to conduct a POC in a written document before beginning a POC.

A.1.4 Expense of POC

Perhaps the most difficult question of all is: who bears the expense? The answer to this depends entirely on the situation. The bottom line is this: a POC will never be a free-of-charge event. Determine ahead of time who will

bear the expense, and make certain this understanding is agreed to by all involved.

Appendix B. Test for Indicators of Performance (TIP)

In Appendix A, “Proof of Concept (POC)” on page 133, we discuss one aspect of validating an e-business design. Another aspect might be to assess the performance of the design.

Assessing the performance of an e-business design that has not yet been fully specified or implemented might appear to be an impossible task. Some level of valuable performance testing can be accomplished, but a Test for Indicators of Performance (TIP) at this point should *not* be viewed as a complete verification of the performance of the final design.

What a TIP can do, as the name implies, is provide *indicators of performance*, which are key measurements that offer some level of prediction of the behavior of the final system. A rough measure of such an indicator would be if one component of your system consumes 80% of your total response time expectation. Such a disproportionate usage might indicate a problem. Another indicator would be the trend exhibited as additional workload is placed on the system. A curve with a dramatic spike in the response time as workload is introduced is not a positive sign.

Before embarking on the expensive TIP route, it may be valuable to consider some lower-cost activities based around modeling.

B.1 Is Performance Modeling an Option

Modeling activity falls into a number of categories, each of which trades increasing sophistication (and hence accuracy) for cost.

1. At the simplest level, simple “rule of thumb” estimating is often very valuable. Straightforward assumptions such as the following should usually be applied as a basic feasibility check before any TIP work is started:

   - All server transactions are 300 msec.
   - Ethernet LANs should not be driven over 60% of capacity.
   - A standard A4 scanned b/w image is 25 kilobytes.
   - A Citrix server should only have 50 clients.
   - A SET transaction will be at least 2 kilobytes, regardless of the business data.

   This approach will also start to indicate any potential bottlenecks in the system, especially if applied to the basic building blocks identified in the (alternative) architectures and to their interconnections.

2. At the next level, some simple analytic modeling using basic queuing theory may be useful. The use of anything other than very simple queuing theory is a complex subject and unless you have a formal understanding of the assumptions underlying the formulae, it is very easy to be badly misled.

   However, for straightforward system models of the type described in this redbook, queuing theory can give some indication of how the loading/utilization of system components can affect their response time (and hence the aggregate end-to-end response of the business transaction). It will also indicate at roughly what business volume you are likely to start to "climb a curve" and encounter real response problems.

   Much of this work can be done with spreadsheets; the use of their "optimization/goal seeking" tools can often help with allocating resources to the overall system (for example, is it better to increase this bandwidth at cost X or add another server at cost Y?).

3. At the most sophisticated level, a full simulation of the system can be performed using a discrete simulation package. This is a significant activity, but may still be cheaper than a TIP. It can also be used where a TIP is not feasible (for example, because the platform is unavailable or has not yet been built).

   This approach requires the characterization of the system components in terms of response time, capacity, queue strategy (for example, FIFO) and connectivity. It also requires some assumptions about the arrival rate and distribution of business transactions which this is often a valuable way of getting the client to think about this subject in the first place.

   It is worth noting that this form of modeling need not be confined to technical aspects; the entire business transaction can be modeled, using "livery" data for the people and business processes involved, and technical data for the platform elements.

   Note that this approach is especially valuable for e-business applications as it allows the simulation of very large loads (for example, "IBM Olympic Web site" hit rates), together with their user keying patterns and so on, as well as other forms of stress testing. These are almost impossible with a TIP.

   The IT Architecture and Design ICM database contains several documents on the use of the SES/workbench modeling tool in client situations. See Appendix F.2.7, "IT Architecture and Design" on page 168 for more details of the database.

B.2 Planning the TIP

If you have run a Proof of Concept, you will have a functional prototype or some part of a prototype against which you can perform the TIP, but you may not have started any development.

The best approach is to identify key measurements you can take during performance testing. The end-to-end measurement may be the response time from click of the mouse at the browser to returned HTML page. For the purposes of a TIP, however, you may want to break that down to the components of the total response time. When you measure the performance of these identified components, you would be looking for things such as:

- Does the response time of this component take up a disproportionate amount of the total end-to-end response time “budget” you have defined for this solution?
- As you drive more and more workload across each component, does the response time degrade? If so, does it degrade in a linear fashion, or does it degrade at an increasing rate?

B.2.1 Preparation for the TIP

Before beginning a TIP, you should spend time developing answers to the following questions:

- What will be tested and how will it be tested?
- What is the degree of customer involvement? Have you gained the customer’s agreement? Who will bear the expense of the TIP?

  These last issues are the same as discussed for the Proof of Concept, starting at A.1.2, “Degree of Customer Involvement in POC” on page 135.

B.2.2 What Will Be Tested, How It Will Be Tested

Measuring the end-to-end response time of your prototype system might not prove much at this point in time. If the response time is within your desired results, you have not really determined the behavior of the individual components. If the response time is inadequate, you cannot determine why that is the case.

By breaking the system down into components, you can analyze them individually, and understand the component behavior more completely. Later, “whole system” performance testing can be accomplished.

B.2.2.1 Break System Down Into Components

To describe this task, take for example the scenario used in Appendix A, “Proof of Concept (POC)” on page 133, shown in Figure 36.

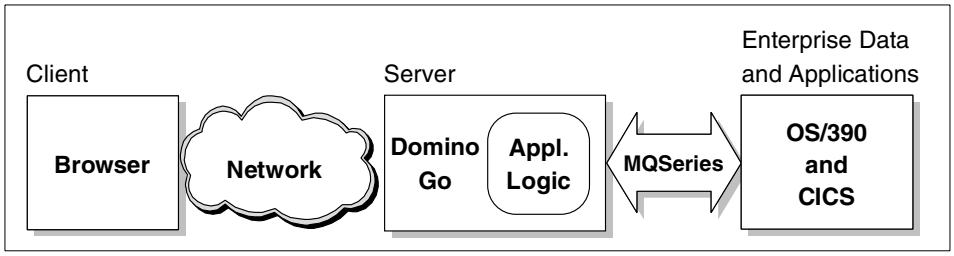


Figure 36. Example of Design Used for Performance Testing

In the pre-sales phase of this solution design, you might very well have a very rudimentary prototype constructed:

- The “application logic” may consist of nothing more than a single defined transaction with a connection to MQSeries.
- The connection from MQSeries to CICS may be for only one defined transaction on CICS
- The network might be only a test floor approximation of the real environment.

Review with the IBM design team and the customer those components of the total system that comprise likely key performance areas. For example, you might come up with a picture that identifies the areas to measure as shown in Figure 37 on page 142.

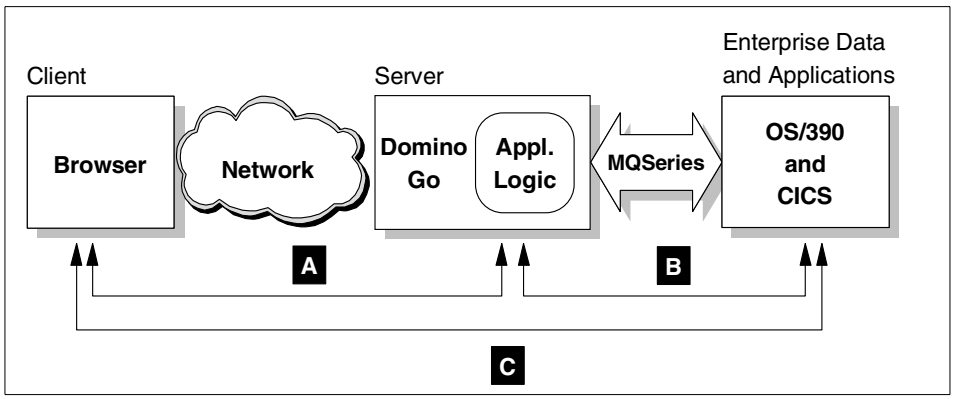


Figure 37. Performance Testing Components of End-to-End Design

A Round trip response time from browser to front of application code

This would measure the response time of your test system from the browser, through the network, through the Web server, and to the very front edge of your application code. For the purposes of this test, you may wish to code a simple “echo” function in your code to remove the behavior of the test code from this test.

Note: If the network in your solution design is an *intranet*, then you have some control over the capacity and behavior of the network, and thus some control over the response time you can assure. If the network is the *Internet*, then the behavior of the network is largely out of your control, and you should be careful to avoid any service level agreements based on response time through the Internet.

B Round trip response time from application code to host system

This would measure the response time from and including the application code, through MQSeries and to the CICS system.

C Overall round trip response time of system

This represents the overall system response time. The expected “response time service levels”, or your “response time budget” maps to this.

B.2.2.2 Determine What You Will Measure

Response time is one measure of performance, and from the user point of view, perhaps the most important. However, as you are testing for response time, you should also consider measuring other system metric, such as CPU utilization or network utilization.

As previously mentioned, be careful about setting any expectations about response time across the Internet. That is a network over which you have very little control. Therefore, if your final solution will operate across the Internet, you may test for performance just to understand the behavior, but avoid any service level agreements based on your testing across the Internet.

B.2.2.3 Determine How You Will Drive the System

A key objective of any performance testing is the observation of the system under load. Measuring a single user’s response time through the system provides an indication of the rough functioning of the system, but will not tell you how the system will perform with 10, 100, 1000 or more concurrent users.

Many tools exist on the market to simulate workload. Based on your system design, choose a tool that will do the job. Then spend some time defining some testing parameters:

- What is the anticipated real user base?
- What will be the levels to which this test system will be driven? You should drive your test system at several times expected load, as well as at measured intervals to see the trend of performance (see B.2.2.4, “Measuring Intervals” on page 144).
- What is the expected measurement result for each component being tested? In other words, if in the example provided in Figure 37 on page 142, testing of component “A” resulted in a response time of.8 seconds, is that acceptable or unacceptable?
- What is the total end-to-end “budget” for response time? This number is useful because it helps compare the results of individual components. For example, if the “budget” is 1.2 seconds and one component uses 1.0 seconds, that might indicate a disproportionate use of the overall “budget”. Therefore, that component might require closer scrutiny.

B.2.2.4 Measuring Intervals

Understanding the trend of a system’s performance across workloads is just as important as understanding the absolute performance of the system. For example, if you measure a system at workload intervals of 100, 200, 300, 400 and 500 users, and you see a curve that looks like that shown in Figure 38 on page 144, then you are probably looking at a relatively well-behaved system, at least within the range of 100 to 500 users.

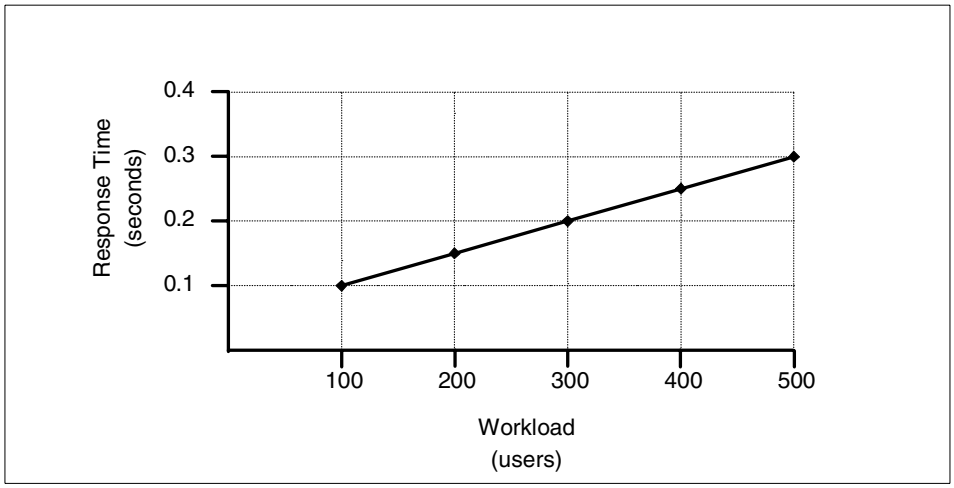


Figure 38. Linear Performance Curve

On the other hand, if your curve looks like that shown in Figure 39, then you would need to investigate that component to understand why such a performance trend was occurring.

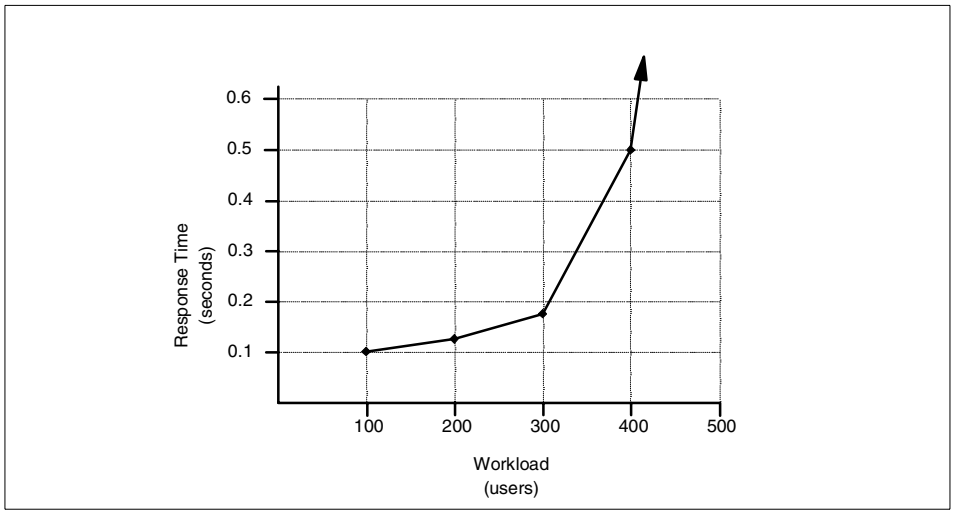


Figure 39. Non-Linear Performance Curve

B.3 Setting Expectations

Be careful to insure expectations are properly set going into a Test for Indicators of Performance. The danger is that if the system performs extraordinarily well, the customer may believe that will be the case with the final implementation. Only testing *after* final solution construction can prove what may be suggested in this assessment test.

B.4 The Lasting Benefits of a TIP

One of the benefits of performing a TIP, given that one is called for, is that once you have developed the “script” used to drive the system, you can use that “script” time and again as you progress through your system implementation. In this way, you can validate your expectations as the system is built, rather than waiting until the very end.

Appendix C. Sample e-business Solution Designs

In order to simplify the process of designing an e-business solution and to reduce the risks, it is often helpful to base your design on a proven system.

There are many sources of sample designs available within IBM. We have listed important sources of information in Appendix H.6, “Intellectual Capital Databases” on page 177. More design guidance is available from some of the intranet sites listed in Appendix H.5, “URLs” on page 176.

This chapter shows you some example solutions taken from the various sources referenced. What is presented here is a high-level overview of the designs, but there is a large amount of detailed supporting material available in the source databases.

We recommend that you refer to the original source databases when looking for sample designs because of the additional level of detail available there and because the information in the database tends to be most current.

C.1 e-business Solution Types

Following a detailed study into e-business commissioned from an external consultant, IBM has been categorizing e-business opportunities into five areas which fall under the following headings:

- Teaming, Collaboration and Knowledge Management
- Enhancing Customer Relationships
- Supply Chain management including Billing and Payment
- Business Intelligence
- e-Commerce

Of course, in practice there are not always clear distinctions. For example, one of the scenarios from the Cross-Platform Integration Test (CPIT) team is entitled “Using Domino to Manage Workflow between Companies over the Web”, which could also be categorized under Collaboration or Supply Chain management if that were the business problem that the application addressed.

C.1.1 Knowledge Management, Workflow and Collaboration

This description and Figure 40 are taken from the ESS pattern “Asynchronous Collaboration’ for Action Requests”.

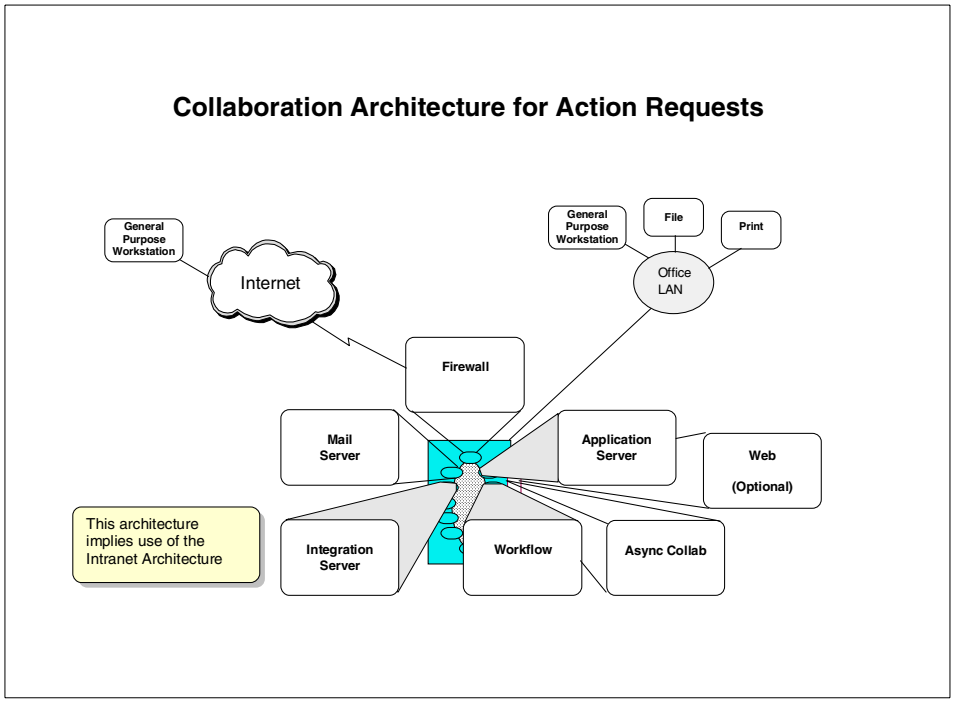


Figure 40. Collaboration Architecture

A workflow can be defined as the completion of a task or tasks through the directed movement of information from an initial state to a final state. These state changes may or may not involve multiple participants. A system that implements a workflow where state changes involve the moving of information from one participant to another can be considered an asynchronous collaborative system. However, some workflow systems involve only a single person (for example, a medical self-diagnosis system, a tax preparation application, or a mortgage preparation application) and therefore cannot be considered a collaborative system.

Additionally, certain asynchronous collaboration styles, such as discussion databases or bulletin boards where information is posted as new topics or responses within a category, do not constitute a workflow since they do not involve the directed movement of information from an initial state to a final state.

C.1.2 Commerce and Personalized Services

In many potential applications of e-business a kind of 80/20 rule applies. That is to say, 80% of the revenue or even profit comes from 20% of the customers. For example, in the case of an airline with a frequent flyer program, the top customers are the ones most likely to have access to Web technology. They are the ones who could exploit a personalized Web-based service designed to keep their loyalty and encourage them to use more of your products or services.

The “added value” provided by the now-famous amazon.com enterprise is due not only to its low prices, but also to its efficiency and ability to suggest books that its registered customers might like to read based on information they supply or on their previous buying habits.

The sample e-commerce environment uses Net.Commerce (Pro Version 3.1) to provide the storefront, along with Net.Commerce's integration capabilities to tie online purchases to an existing CICS inventory application, using MQSeries. Scalability is provided by using two Windows NT commerce servers, load balanced with IBM eNetwork Dispatcher running on an AIX platform, to handle the Internet shopping requests. For security, a screened subnet architecture using two firewalls is shown: one on Windows NT and one on AIX. To respond to Web clients shopping on Windows 95, Windows NT and OS/2 platforms, the Windows NT commerce servers interact with a commerce database on an AIX platform and the inventory database on an OS/390 platform.

This design and Figure 41 are taken from the Product Integration Test Team report “Online Catalog to Enterprise Data in a Secure Environment “. See

http://www.software.ibm.com/ebusiness/cpit2.html for a detailed description.

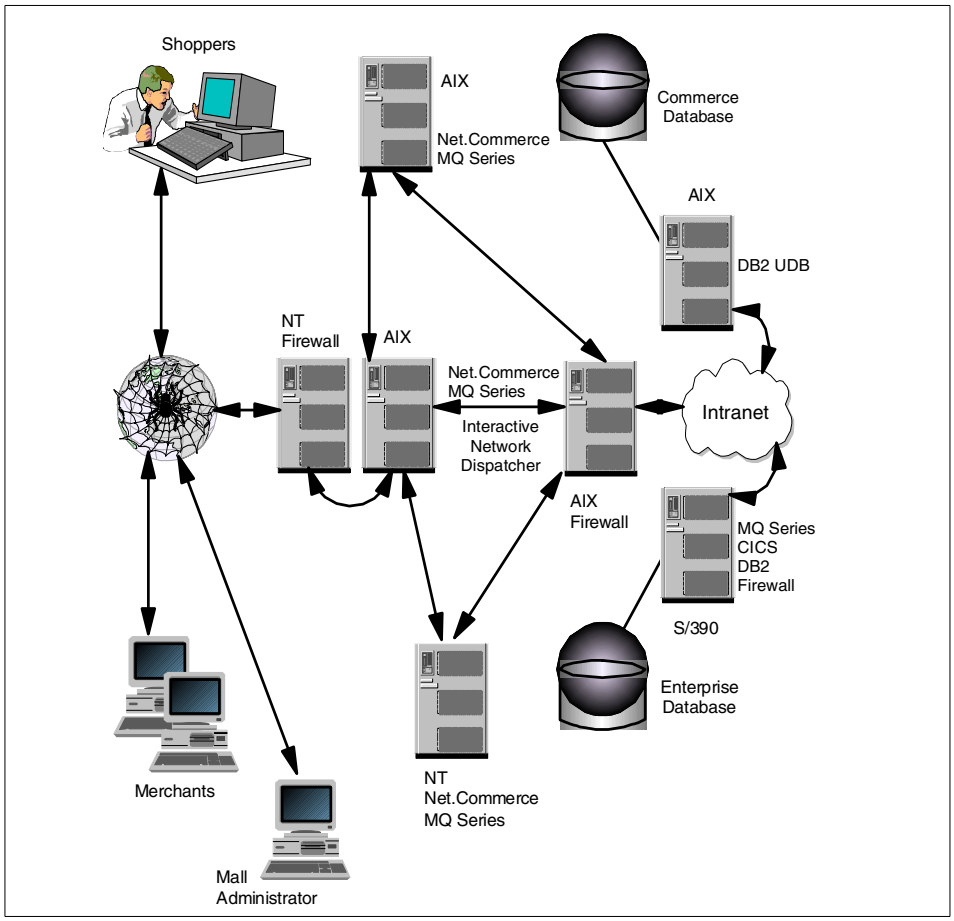


Figure 41. e-commerce

C.1.3 Supply Chain

The example in Figure 42 is taken from the Product Integration Test Team report on “Connecting CICS and DB2 Enterprise Data to The Web”. See http://www.software.ibm.com/ebusiness/ecics.html for more details.

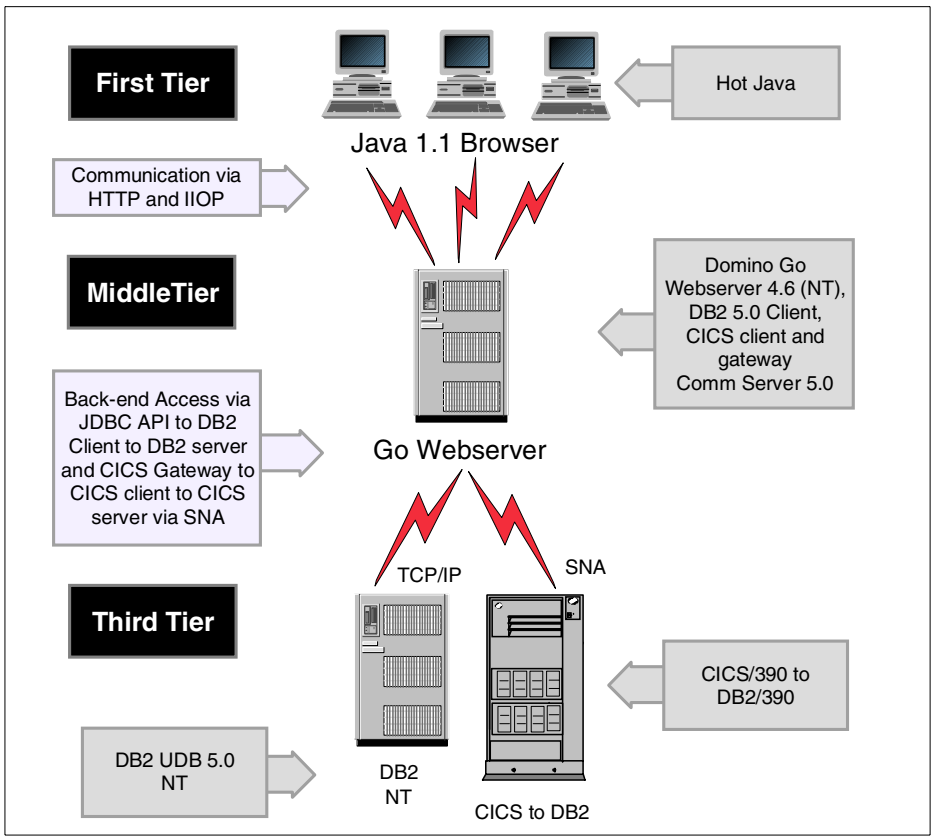


Figure 42. Supply Chain

This design is based on an insurance company interested in improving its business by extending it to the Web. The application provides agents the ability to view customer insurance policy data through a Web browser within the company’s intranet. The policy data resides in an existing DB2 database. Additional data resides on a CICS-based, OLTP system that uses DB2 as the relational datastore.

C.1.4 Customer Relationships

This description and Figure 43 are taken from ESS.

This Operational Architecture describes how an IT System can be structured to provide access to an enterprise's core transaction systems from the Internet. The Transactional Base operational architecture is used to describe the core systems and the TC Internet architecture is used to describe access from the Internet.

The key points displayed by the architecture are as follows:

- Network security zones: Web servers are in a separate zone of the network between the public Internet and the enterprise's core internal network. This allows appropriate security controls to be implemented for each zone.
- Separation of informational Web service from transactional Web service due to the difference in non-functional requirements. This allows an open, easily accessible public information service and a secure transactional service.
- No direct connection from the Internet to core transactional systems.

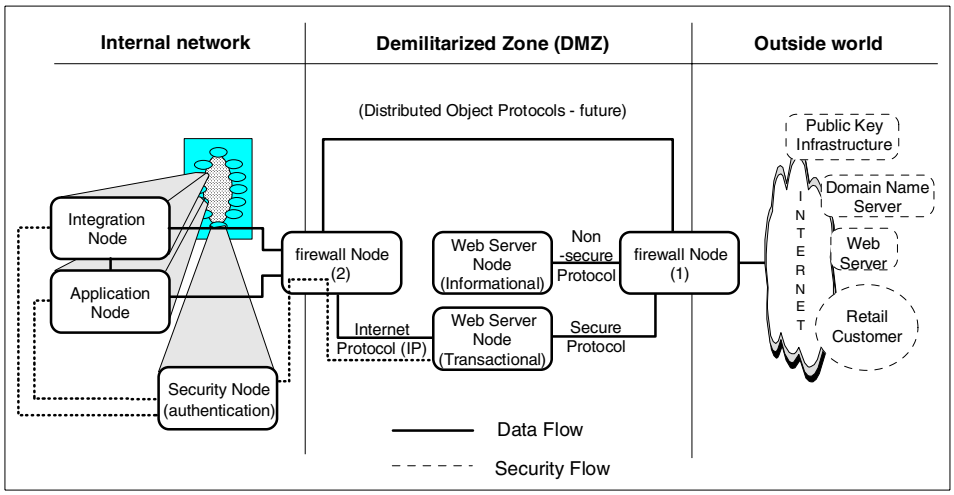


Figure 43. Internet Access to Corporate Applications

C.1.5 Business Intelligence

The term 'Business Intelligence' implies the exploitation of corporate data assets for competitive advantage. Business Intelligence applications include Executive Information Systems, Decision Support Systems and Data Mining applications.

Although Business Intelligence was identified as one of the opportunity areas for e-business, it is a specialized area and you should seek the assistance of subject matter experts.

The public area of the Business Intelligence ICM Knowledge Network might be a good place to start research. See Appendix F.2, “ICM AssetWeb” on page 165 for information on how to request access to the ICM AssetWeb.

Appendix D. The WebSphere Family of Products

The IBM WebSphere Application Server is the result of the evolution of what was traditionally the “Web” server. The addition of the “application” acknowledges the fact that this server is no longer simply serving HTML, but also industry-strength business applications. In some ways it is also the gateway to data and applications on back-end, third-tier systems. A large number of applications on the Web server are simply gateways to an existing back-end application or server, and use a set of “connectors” for access to this back end.

With the announcement of WebSphere, IBM has consolidated many different product offerings under one packaging scheme that simplifies the selection of products to support an e-business design. The following three offerings of the WebSphere Application Server allow a customer to begin a migration to e-business with simple content hosting, and later move up to full mission-critical enterprise applications:

1. WebSphere Application Server, Standard Edition
2. WebSphere Application Server, Advanced Edition
3. WebSphere Application Server, Enterprise Edition

Associated products that support the WebSphere Application Server environment include WebSphere Studio (see Appendix E, “IBM Application Development Tools Classification” on page 161) and WebSphere Performance Pack (see Appendix D.4, “WebSphere Performance Pack” on page 157. Together, the WebSphere Application Server family (as represented in Figure 44 on page 156) provides IBM customers with a comprehensive environment in which to develop, build and deploy an e-business solution.

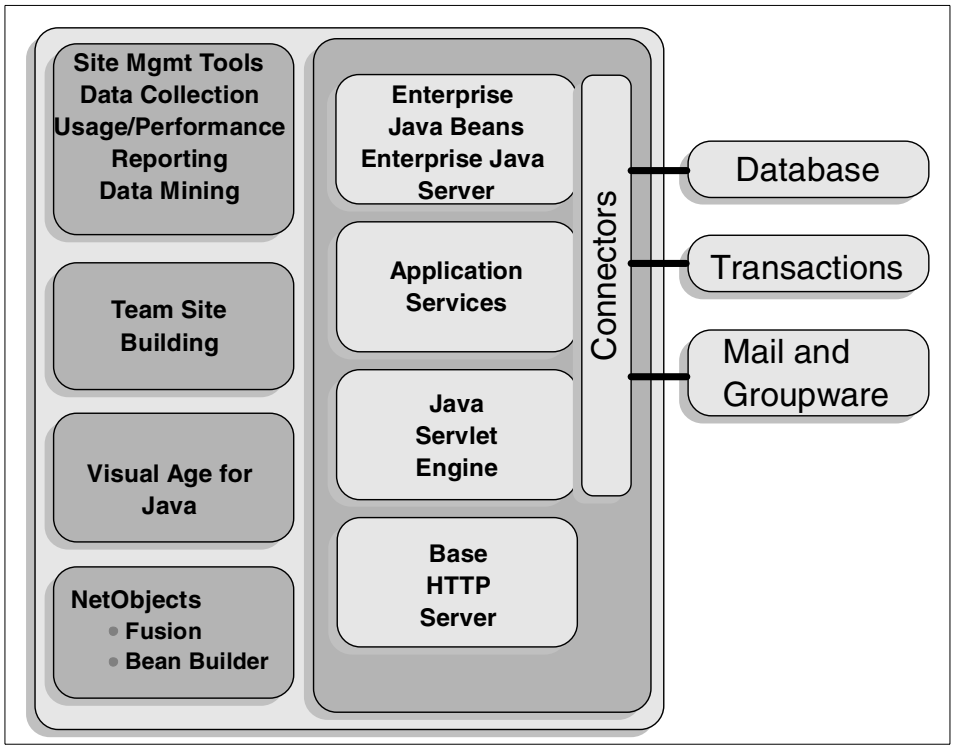


Figure 44. WebSphere Application Server

D.1 WebSphere Application Server, Standard Edition

WebSphere Application Server, Standard Edition forms the foundation of the entire application server family and offers customers a solution to build, deploy and manage e-business Web sites. The Standard edition provides companies with an open, standards-based, Web server deployment platform, as well as Web site development and management tools to help accelerate the process of moving to e-business. Enhanced integration with the IBM DB2 Universal Database V5.2 allows developers to design a wide range of e-business applications.

Standard Edition provides the Web site development and management tools to customers who are beginning the construction and deployment of Web site content. WebSphere Application Server runs on top of the most commonly used HTTP servers, including Apache, Netscape, Microsoft and Domino Go. If a customer does not already have an HTTP server, and depending on your

platform, the WebSphere Application Server, Standard Edition comes with the popular Apache HTTP server and/or another base HTTP server.

D.2 WebSphere Application Server, Advanced Edition

WebSphere Application Server, Advanced Edition builds on the Standard Edition to provide enhanced support for scaling Web sites into secure, transactional e-business applications. The Advanced edition connects Web applications to existing databases and host-based transaction systems, and offers sophisticated tools to simplify distributed component-based application development. This is accomplished by including support for the deployment of Enterprise Java Beans (EJB), e-business connectors and the ability to cluster servers for additional capacity.

D.3 WebSphere Application Server, Enterprise Edition

WebSphere Application Server, Enterprise Edition enhances the Advanced edition and offers a robust solution that grows e-business applications into mission-critical enterprise environments, leveraging existing skills and IT systems. It combines TXSeries, IBM's world-class transactional application environment, with the full distributed object and business process integration capabilities of Component Broker. All applications currently running on TXSeries or Component Broker will be fully supported with the Enterprise edition.

D.4 WebSphere Performance Pack

WebSphere Performance Pack is an integrated Web infrastructure offering that provides caching, file replication and load balancing features, as well as administrative tools. It enables customers to integrate Web technologies with their existing core IT systems to achieve a competitive e-business advantage. WebSphere Performance Pack combines proven technologies in an open, standards-based infrastructure.

The customers who will benefit from the full range of capabilities are Internet Service Providers (ISPs) and Web site managers who need to address problems of server availability, congestion, line costs, and/or file administrative costs in order to meet the expectations and satisfaction criteria of their respective customers.

WebSphere Performance Pack is composed of three major components: an enterprise file system, a caching proxy server and a load balancing/server

monitoring tool. These features provide a flexible combination of caching, proxy and filtering functions, file content management and replication, and load balancing into a single Internet hosting infrastructure.

D.5 Differences Between WebSphere Application Server and Domino

Lotus Domino and IBM's WebSphere Application Server family are robust Web application servers that meet different customer needs. Lotus Domino is ideal for customers who are focusing on collaborative/workflow solutions and want to build intranet and extranet Web applications that integrate their business processes with those IT systems. For example, Chrysler's new supplier suggestion system is built using Lotus Domino and allows suppliers, via an extranet, to send in documents with their suggestions, which are then processed in a workflow system to generate cost savings for the company.

The IBM WebSphere Application Server family is ideal for organizations building more transactional Web applications using Java servlets and Enterprise JavaBeans. The WebSphere Application Servers provide a deployment and management platform that enables companies to upgrade from a publishing-based Web presence on HTTP servers to full transactional e-business solutions.

For example, WebSphere Application Server is the foundation on which the next release of IBM's Net.Commerce will provide e-commerce capabilities that allow customers to set up electronic storefronts with a more personalized, dynamic selling environment.

D.6 When to Use WebSphere Application Server or Domino

The Web Application Server market is a large, loosely defined market. There is significant opportunity at two different ends of the market. At one end, there is a customer need for a transactional Web application server solution. The answer for that need is the WebSphere family.

On the other end, there is a customer need for a Web application server that delivers business process applications and collaborative solutions. The answer for that need is Domino.

D.7 Summary

There are four key features of the WebSphere Application family:

1. A family of compatible, scaleable Web-enabled application servers that range from low-end publishing and dynamic content-oriented sites through high-end application-intensive sites.
2. A high level of commonality between these application servers; they are interoperable and share common infrastructure, administration, and programming model. This allows customers to start at the right level of capability, and easily add or migrate to other members of the family to exploit additional capabilities.
3. The Enterprise JavaBeans programming model provides a simple, powerful component-based paradigm for building business logic.
4. Preservation of the customers' investment in IBM's application servers; including WebSphere Application Server, TXSeries, Component Broker, CICS/390, IMS, and others; through support for Enterprise JavaBeans as well as a range of powerful connectors to existing applications and systems.

These features enable customers, whether starting from Web publishing and moving up to dynamic content and Web applications, or starting with enterprise IT applications and extending them out to the Web to start with the right application server, preserve their investment in existing products, and easily incorporate new capabilities as their applications evolve.

Appendix E. IBM Application Development Tools Classification

It is important to understand the capabilities of the tools provided by IBM when proposing products to meet the customer's application development needs. One way of categorizing the tools is shown in Figure 45.

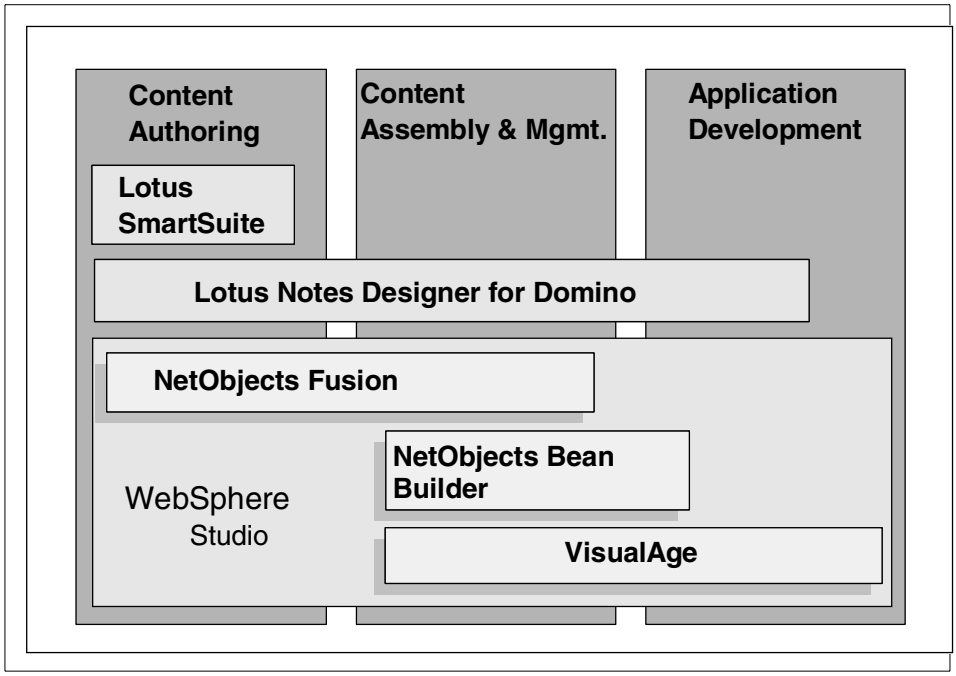


Figure 45. IBM Tool Classification

With the announcement of WebSphere Studio, IBM has made the functions of NetObjects Fusion, NetObjects Bean Builder and VisualAge available in one package. WebSphere Studio 1.0 provides:

- Servlet Generation Wizards: These allow Web site developers who are not familiar with Java, or developers who just want to save time, to generate complex business logic in the form of servlets which run on the WebSphere Application Server. The wizards use interactive prompting to walk users through the process of creating a servlet. Wizards can generate SQL database queries, or produce a servlet from any JavaBean.
- Web Development Workbench: This tool gives you a view of all the components in your Web site projects and allows you to quickly navigate among these components, modifying or viewing them as needed. It provides a helpful Web site project Organizerâ, and also acts as a launch

platform for the tools of choice (such as HTML or graphic editors) for different types of components. Developers can use it to quickly publish entire Web applications to the WebSphere Application Server.

- NetObjects ScriptBuilder: ScriptBuilder helps you do text-based editing on a wide variety of scripting, programming, and markup languages that are used to build Web applications, including Java, JavaScript, JScript, HTML, D-HTML, and JavaServer pages extensions. Its features make the scripting process easy, such as the ability to instantly preview scripted Web pages, smoothly reference and add language elements, and quickly navigate to embedded functions and objects.
- NetObjects Fusion 3.0: Fusion is a powerful Web site creation tool that allows you to visually create a Web site complete with multiple interconnected Web pages, imagery and other media, and dynamic HTML features for interactivity. You can edit the individual pages in a Web site, preview an unpublished version, or view the entire site structure. All the pages in a site can share a common visual appearance via a SiteStyle, which can be quickly changed, modifying the appearance of every page in the site automatically. Note that NetObjects Fusion is also shipped with Domino Designer.
- NetObjects BeanBuilder 1.0: BeanBuilder lets you create Java applets and beans quickly and visually, without Java programming. Applets and beans can be visually assembled on-screen by using a provided set of JavaBean components or by importing any standard JavaBean. Logic can be added allowing the components to interact in the form of “connections” which are simple rules that describe how an event associated with one bean can trigger an action in another bean.
- VisualAge for Java, Profession Edition 2.0: This is provided for Java programmers who want to customize Web applications with custom-created Java components. VisualAge for Java is a robust, fully functional Java programming environment. It allows Java programmers to perform advanced functions, such as incremental compilation and invocation of methods during the debug phase of development.

This packaging simplifies the IBM development tool selection. If the business requirement indicated a strong need for collaborative applications, then the choice is Lotus Domino Designer, with the addition of Java development tools if needed. Otherwise, WebSphere Studio is a comprehensive solution.

Appendix F. Other Design Methods and Sources of Information

Over the past few years, many different design and development methods have been devised and used within IBM, and even more methods exist outside IBM.

This appendix provides a brief description of the main IBM design methods with which the authors are familiar. Appendix F.2, “ICM AssetWeb” on page 165 describes the Intellectual Capital Management AssetWeb system, tells you how to request access to it, and lists those of its databases that are most relevant to e-business.

F.1 Methods and Sources of Information

IBM has a wealth of intellectual capital based on successful results of prior customer engagements. This intellectual capital is used and updated by IBM skilled practitioners, and provides many proven design methods and patterns that can be used to address all or parts of an e-business solution. This section describes the most widely used design methods and patterns.

F.1.1 SI

Systems Integration (formerly SI/AD) is one of the six competency areas of IBM Global Services. Members of the SI competency are currently organized into four competency segment groups:

1. Custom Application Development
2. Application Development Maintenance
3. Application Enabling Integration
4. Package Integration

You can find a definition of the Global Services competencies at the following IBM intranet site: http://w3.ibm.com/services/competencies/

F.1.2 WSDDM

WorldWide Solution Design and Delivery Methods (WSDDM) is a collection of generic plans and methods representing IBM's best practices for planning, managing, and delivering projects. WSDDM is based on the cumulative knowledge compiled from hundreds of successful engagements and projects. WSDDM methods also include detailed Object Technology and Project Management methods. The WSDDM home page is at:
http://w3.wsddm.ibm.com/

WSDDM is now classed as one of the methods within the Application Development Maintenance (ADM) segment of SI. For a description and brief history of WSDDM see:
http://w3.ibm.com/services/resources/delguid4.html

F.1.3 AD/SI

This refers to the Application Development/Systems Integration (AD/SI) Method Group within WSDDM. AD/SI is a collection of nine design methods, including Infrastructure Design.

For more information go to:
http://sbywsddm.somers.hqregion.ibm.com/
(and select **Solution Delivery**)

F.1.4 ISD

Infrastructure Design (ISD) is a path of the AD/SI method component of WSDDM. ISD is used to design the delivery platform supporting an application. ISD phases are Requirements, Architecture, Infrastructure Specification and Component Specification and Selection. ISD is based on the original E2E method, which provided a formal way to design complex systems.

Readers familiar with ISD (or E2E) will notice similarities to some of the questions in this redbook. The formal ISD method covers many of the same design and system management issues, in detail. For further information about ISD, refer to the AD/SI site:
http://sbywsddm.somers.hqregion.ibm.com/

F.1.5 CAS

The Complex Architecture Solutions (CAS) methodology is used to guide the design of an end-to-end solution for a specific business objective. In addition, a thorough understanding of the target or client environment is generated. The phases defined by CAS are: Document the Environment, Define Architectural Building Blocks, Define Integrated Solution, Evaluate Integrated Solution, Create Solution Presentation and Close Out.

The output from the Close Out phase of CAS should be a design to be handed over to the delivery team.

The CAS methodology is highly structured and calls for many well-defined documents to be delivered at each step of the process. CAS is designed to make heavy use of a Notes-based Solution Builder tool. You can read more

about the Solution Builder by starting at:
http://w3.ncs.ibm.com/cas/main+navigator.html.

On this site you will also find much useful reference material and descriptions of standard solutions such as SAP R/3 for OS/390.

The CAS homepage is at:
http://w3.ncs.ibm.com/home.nsf/complex

F.1.6 AVM

The Accelerated Value Method (AVM) is a rapid deployment method, based on iterative prototyping, employed by Lotus Consulting. AVM stresses the importance of focussing on business benefits. It plans for successive prototypes in short, bounded cycles (“Value Frames”), in order to allow clients to start to realize business benefits as soon as possible.

Five modules are defined within AVM: Engagement Management, Transformation Management, Process Innovation, Collaborative Development and Enterprise Deployment

Although AVM is optimized for Notes/Domino projects, many of the steps defined in the AVM Enterprise Deployment Guidebook will sound familiar to IBM designers and architects. The steps include:

- Business Drivers and Requirements
- Scope of Deployment
- High Level Deployment Plan
- Define Success Criteria
- Skill Assessment
- Existing Notes Environment
- Iterative Deployment
- Security Policies
- Migration and Coexistence Strategy

Find out more about AVM at the Lotus Consulting site:
http://www.lotus.com/home.nsf/welcome/consulting
(and select **Methodology**).

F.2 ICM AssetWeb

The ICM AssetWeb was created to allow consultants and other IBM services professionals from around the globe to share methodologies, tools and other intellectual capital. In March 1998 it won the gold award in the category of document and knowledge management at the Ninth Annual Giga Excellence

Awards against competition from other leading players in the industry. See: http://domino.www.ibm.com/Services/pressrel/pr.889566539.html

The ICM AssetWeb system has, at the time of writing, about 50 knowledge network databases, replicated on Domino servers around the world, available for use by Global Services and related professionals on client engagements.

The ICM databases are front-ended by the AssetWeb Navigator database which is open to access by anyone with a /IBM or /Tivoli Notes ID. Most other ICM databases require authorization.

Instructions for accessing the AssetWeb Navigator are on the IBM intranet at: http://w3.ibm.com/services/smns/htuicm.html

The ICM Navigator contains a form with which users can request access to ICM databases after agreeing to ICM terms and conditions. With this authorization, users should be able to access the public areas of ICM databases. Each ICM database usually also contains an area which is restricted to practitioners trained in the methods of the database.

Use of ICM material

Be sure to read the Legal and Ethical Guidelines which are accessible from the front page of the AssetWeb Navigator. There you will find advice on protecting our clients' information and IBM's Intellectual Capital.

F.2.1 e-business Knowledge Network

The mission of the e-business knowledge network is to provide an integrated set of knowledge, experience, analytics, tools and techniques which IBM practitioners can use to determine and deliver effective and efficient client e-business solutions. This includes business, organizational, application and technical considerations and frameworks which will be supportive of client needs, opportunities and objectives.

The competency defines the following stages of an e-business project: Engagement Marketing, Strategy, Plan, Architect, Construct and Operate. The database includes advice and guidance for all these phases. It also describes several e-business service offerings.

F.2.2 Enterprise Solutions Structure (ESS)

The Enterprise Solutions Structure (ESS) builds on the experience gained through E2E and enhances it through the concept of reusable standard

solutions or “patterns”, which are designed to be easily adaptable to new applications.

In the ESS ICM database you can find a pointer to the ESS Tool database, in which you will find sample architecture definitions that may be useful starting points for designs.

Additional information about ESS can be found in the *IBM Systems Journal Volume 38, Number 1, 1999*, which provides information about technical reference architectures. It also includes the Thin Client Transactional Model from the context of the Enterprise Solutions Structure (ESS) Project. Articles cover the following topics:

- Introduction to ESS (E. Plachy and P. Hausler)
- A Standard for Architecture Description (R. Youngs)
- A Standard for Business Architecture Description (D. McDavid)
- Solution Customization (D. Leishman)
- Technical Reference Architectures (P. T. L. Lloyd and G. Galambos)
- Experiences in Reusing Technical Reference Architectures (T. Harris)

F.2.3 IT Security

The IT Security ICM database, in its public area, contains much information of direct relevance to the design of secure Internet applications. The many subjects covered include Firewalls, Identification, Authentication, Confidentiality, Privacy and Disaster Recovery.

F.2.4 Intranet/Extranet Offerings

Categories in the public area of this database include Enterprise Messaging, and Infrastructure Assessment, Architecture Design, Best Practices, and Implementation. As well as guidance in the various categories, the database contains sample project definitions (Statements of Work).

F.2.5 Microsoft Technology

The Microsoft Technology database is intended to extend beyond the Microsoft Services competency and connect all of IBM's Microsoft-skilled professionals. The categories in this database that are highly relevant to e-business include: Groupware and Messaging, Database Software, Web Software, Middleware, Systems Management, Security and Directory Services, Architecture and Design, and Tools and Languages.

F.2.6 Network Architecture and Design

The public area of this ICM database covers Network Assessment, Architecture, Design, Implementation and Management. It also includes guidance on Engagement marketing.

F.2.7 IT Architecture and Design

The IT Architecture Knowledge Network database covers architectures for data, applications, information, systems management, application development and technology (platforms, communications, and system services). Its methods are IT Architecture (ITA) definition and ISD/E2E. The ISD/E2E method can be applied to the design of e-business solutions, although few examples in this ICM database are explicitly linked to e-business.

More importantly, the performance techniques section of this ICM database contains several documents offering advice and guidance about performance estimation and modeling tools. You could use the guidelines from this database to assess whether to use estimating and modeling tools, or the TIP process described in Appendix B, “Test for Indicators of Performance (TIP)” on page 139, or both.

F.2.8 Small and Medium Businesses

This database is intended to be a repository of useful assets for those working with small and medium business. It contains sample proposals for an Internet/Intranet Business Plan and Notes/Internet workflow.

F.2.9 Lotus Notes

At the time of writing the Lotus Notes ICM database is still listed on the Asset Web Navigator panel. However, in the future it will be replaced by the e-business ICM database, which will include Lotus Notes support, among the other topics.

Appendix G. Special Notices

This publication is intended to help IBM technical professionals who are involved in the pre-sales opportunity cycle of e-business solutions. The information in this publication is not intended as the specification of any programming interfaces that are discussed in this document. See the PUBLICATIONS section of the IBM Programming Announcement for any IBM products mentioned in this document for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have

been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain 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.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

You can reproduce a page in this document as a transparency, if that page has the copyright notice on it. The copyright notice must appear on each page being reproduced.

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

AIX	CICS
DB2	DB2 Universal Database
eNetwork	IBM
MQ	MQSeries
Netfinity	OpenEdition
390	OS/390
400	System/360
TXSeries	VisualAge
VTAM	WebSphere

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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

Appendix H. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

H.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 179.

Table 38 on page 173 lists redbooks that are specifically recommended for further reading about aspects of e-business solutions. Visit the ITSO Web sites `http://www.redbooks.ibm.com/` or `http://w3.itso.ibm.com` on the IBM intranet, for an up-to-date list of redbooks in print and in preparation.

Table 38. e-business Redbooks

Title	Order Number
IBM Network Computing Framework for e-business Guide	SG24-2119
Lotus Solutions for the Enterprise, Volume 5: NotesPump, The Enterprise Data Mover	SG24-5255
Enterprise Integration with Domino for S/390	SG24-5150
From Client/Server to Network Computing: A Migration to Domino	SG24-5087

H.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040

CD-ROM Title	Subscription Number	Collection Kit Number
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

H.3 Other Publications

These publications are also relevant as further information sources:

Table 39 gives a very small sample of external publications that may also be relevant as further information sources. Several of these books are by IBM authors and most are orderable using the IBM internal puborder process:

Table 39. External Publications

Title and Author(s)	Publisher	ISBN
Designing Systems for Internet Commerce by G. Winfield Treese and Lawrence C. Stewart 374 pages, (May 98) SR23-8530	Addison-Wesley	0201571676
Firewalls and Internet Security: Repelling the Wily Hacker by William R. Cheswick, Steven M. Bellovin 306 pages, 2nd edition (February 1999)	Addison-Wesley	020163466X
Practical UNIX and Internet Security by Simson Garfinkel and Gene Spafford 1004 pages, 2nd edition (April 1996) SR23-7467	O'Reilly & Associates	1565921488
Java Network Security (The ITSO Networking Series) by Robert Macgregor (Editor), Dave Durbin, John Owlett, and Andrew Yeomans 232 pages, Book and CD-ROM edition (January 1998) SG24-2109	Prentice Hall	0137615299
Capacity Planning For Web Performance: Metrics, Models, and Methods by Daniel A. Menasce, George Mason University, Virgilio A. F. Almeida, Federal University of Minas Gerais, Brazil 450 pages, Book and CD-ROM edition (June 1998)	Prentice Hall	0136938221
Lotus Notes and Domino Network Design by John P. Lamb and Peter W. Lew 575 pages, Book and CD-ROM edition (September 1997) SR23-8120	McGraw-Hill	0079132413

Title and Author(s)	Publisher	ISBN
Lotus Notes & Domino 5 Scalable Network Design With CD-ROM by John P. Lamb 640 pages, (January 1999)	McGraw-Hill	007913792X
Web Gateway Tools: Connecting IBM and Lotus Applications to the Web by Cheng and Malaika 442 pages, (March 1997) SR23-7862	John Wiley & Sons	0471175552
Building Intranets With Lotus Notes & Domino 5.0: How to Provide Your Employees and Customers With Instant Access to the Information They Need by Steve Krantz, Eileen Rudden 290 pages, 2nd edition (August 1998) S246-0162 Comes with password to access companion Web site	Maximum Press	1885068247
Exploring IBM's Bold Internet Strategy: An Inside Look at Big Blue's Vision for the Future of Computing by Jill Ellsworth, Vince Lupiano, Ernest Evans 250 pages, 2nd edition (1999) (First edition was S246-0163)	Maximum Press	1885068239
Marketing on the Internet: A 7-Step Plan for Selling Your Products, Services, and Image to Millions Over the Internet by Jan Zimmerman, Michael Mathiesen, Jerry Yanj 270 pages, 3rd edition (May 1998) S246-0109	Maximum Press	1885068263
Business-To-Business Internet Marketing: 5 Proven Strategies for Increasing Profits Through Internet Direct Marketing by Barry Silverstein 398 pages, (October 1998) S246-0165	Maximum Press	1885068352
Exploring IBM's New Age Mainframes: See Why IBM's Re-Designed S/390 Computer Family Is More Popular Than Ever by Jim Hoskins, Jim Fletcher 400 pages, 6th edition (January 1999) (Fifth edition was G326-3006)	Maximum Press	1885068301

H.4 Other Resources

- The Transaction Systems e-business Solution Kit is orderable from DEMOcentral at `http://w3.demopkg.ibm.com`, and described under

"Application Development". Packaged on four CDs at the time of writing, the kit integrates UDB, LotusNotes and Domino, MQSeries, TXSeries, VisualAge Java, VisualBasic, Powerbuilder, Interspace(tm), VisualAge C++, and VisualAge COBOL into a package that you can use for rapid demonstrations or prototyping.

H.5 URLs

Table 40 gives a list, in no particular order, of IBM sites on the Internet and intranet. All of these references are relevant to aspects of e-business solutions.

Table 40. IBM e-business-Related URLs

URL	Title and Comments
http://www.ibm.com/e-business/	IBM e-business home page.
http://advisor.internet.ibm.com/randr	IBM's Risk and Readiness Advisor.
http://www7.software.ibm.com/vad.nsf/	Visual Age Developer Domain: respiratory for Java Beans.
http://w3.ncs.ibm.com	e-business Advisor.
http://www.software.ibm.com/net.media/	IBM Internet Media page - solutions to add rich media content to sites.
http://w3.ncs.ibm.com/knowledgebase.nsf/	e-business Value Knowledgebase Contains case studies and advice on selling strategies based on independent research.
http://w3.ncs.ibm.com/pcid.nsf/Industry	e-business Technical References by Industry.
http://w3.ncs.ibm.com/home.nsf/competition	IBM's e-business competitors.
http://w3.ncs.ibm.com/e-biz.nsf/VC/focus+offerings	e-business focused offerings.
http://w3.ibm.com/services/smns/htuicm.html	Instructions on how to get access to ICM AssetWeb databases.
http://w3.mfg.ibm.com/mfg/eb/	e-business for the Manufacturing Industries.
http://sbywsddm.somers.hqregion.ibm.com http://w3.wsddm.ibm.com/	Worldwide Solution Design and Delivery Methodology Web sites.
http://www.software.ibm.com/ebusiness	Network Computing for e-business.
http://www.software.ibm.com/ebusiness/buzz.html	Product Integration Test team reports.

URL	Title and Comments
http://w3.ibm.com/services/institute/e-business/default.html	IBM Global Services e-business Information Resource.
http://www.ibm.com/security/html/cryptography.html	IBM SecureWay Feature Story on Cryptography.
http://www.ibm.com/security/html/pr_pkixpr.html	IBM and Lotus Deliver PKIX Software to MIT. The PKIX reference implementation provides a common security infrastructure for the use and management of public-key cryptography and certificates, including key, certificate, and policy management.
http://w3.Lotus.com/intranet/home3.nsf/framesets/ebiz	Lotus e-business intranet site.
http://www.efuse.com/	NetObjects Internet home page.
http://w3.ncs.ibm.com/home.nsf/complex	Complex Architecture Solutions intranet site.
http://www.s390.ibm.com/nc/	IBM S/390 e-business site.
http://w3.software.ibm.com/websphere	IBM WebSphere intranet home page.
http://www.ibm.com/websphere	IBM WebSphere Internet home page.
http://w3.education.ibm.com	IBM Education intranet home page.
http://www.training.ibm.com	IBM Education and Training Internet home page.
http://w3.osc.ibm.com	IBM Global e-business Integration providing Solution Workshops, Design Services, Proof of Concepts and Performance Assessment Testing.
Notes: 1 If you are viewing the PDF or HTML version of this document online, you should be able to click these URLs to open the links in your browser. 2 The URLs were correct at the time of writing. If they have changed by the time you read this, we recommend you use a search engine to find their new location. 3 The convention is that addresses starting with http://w3. are on the IBM intranet. Since this document is aimed at an IBM internal audience we have not distinguished between internal and external addresses in this table.	

H.6 Intellectual Capital Databases

See Appendix F.2, “ICM AssetWeb” on page 165 for a list of ICM databases that contain e-business content.

H.7 Education and Training

Course Name	Course Number
Advanced NCF for e-business	N3106
e-business Solutions with IBM WebSphere	E4400
IBM WebSphere Performance Pack Technical Workshop	E4410
Bridging Legacy Systems to the Web	N3409
Enterprise Java Beans and Component Broker Connector in the WebSphere Environment	ITSS1
e-business Fundamentals	N3102
Essentials of Java for e-business	N3145
An Introduction to Object Technology for Technical Professionals	N1606E

Note: These courses were scheduled in the USA at the time of writing. Refer to IBM Education and Training's Global Campus (http://w3.education.ibm.com) for current schedules and additional courses that might prove helpful to you in designing e-business solutions.

The IBM Global Services Institute home page is at:
http://w3.ibm.com/services/institute/e-business/default.html
On this site you can find information about e-business-related training available from the Global Services Institute. The material includes courseware and presentations, as well as a pointer to the IBM Global Campus.

In EMEA, refer to the La Hulpe International Education Centre site at http://w3.lahulpe.ibm.com to find details of their course schedule.

Also in EMEA, the Installation Support Center (ISC) provides early education on new technologies, including e-business. Visit their site at http://w3.isc.uk.ibm.com for details of upcoming workshops.

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at `http://www.redbooks.ibm.com/`.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Redbooks Web Site on the World Wide Web**

  `http://w3.itso.ibm.com/`

- **PUBORDER** – to order hardcopies in the United States
- **Tools Disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLCAT REDPRINT
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type the following command:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
  ```

- **REDBOOKS Category on INEWS**
- **Online** – send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** – send orders to:

	IBMMAIL	**Internet**
In United States	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** – send orders to:

IBM Publications Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark

- **Fax** – send orders to:

United States (toll free)	1-800-445-9269
Canada	1-800-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 408 256 5422 (Outside USA)** – ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number _____

☐ Credit card number _____

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Glossary

ACL. Access Control List.

AD/SI. IBM's Application Development/Systems Integration Method Group within WSDDM.

API. Application Programming Interface.

APPC. Advanced Program-to-Program Communication.

ASCII. American Standard Code for Information Interchange. The 8-bit character encoding scheme used by most PCs and UNIX systems. It supersedes an earlier 7-bit ASCII standard.

ASP. Microsoft's Active Server Pages.

AVM. Accelerated Value Method. A rapid deployment method, based on iterative prototyping, used by Lotus Consulting.

Bamba. Bamba is a brand name for IBM technology used to develop network-enabled multimedia applications. One function of the technology is streaming, both live and stored, audio and video across the Internet and over intranets via modem or LAN connections.

BB. Building Block. Many of IBM's design and development methods, including the one described in this redbook, construct a design from "building blocks" which are generally defined in more detail as the design evolves.

Bean. A small component that can be used to build applications. See JavaBean.

CAS. IBM's Complex Architecture Solutions methodology. It is used to guide the design of an end-to-end solution for a specific business objective.

CB. IBM's Component Broker.

CGI. Common Gateway Interface.

CICS. Customer Information Control System.

CIG. CICS Internet Gateway.

COBOL. Common Business Oriented Language.

COM. Microsoft's Component Object Model.

CORBA. Common Object Request Broker Architecture.

DB2. IBM Relational Database Family.

DCOM. Microsoft's Distributed Object Model.

DECS. Domino Enterprise Connection Services. A standard component of Domino 4.6.3 and Domino 5 which provides connectivity from Domino to back-end relational data base servers. This connectivity is table-driven and requires no programming.

DMZ. DeMilitarized Zone. This term is now commonly used in the industry to describe a subnetwork, typically used for web servers that are protected by firewalls from both the external Internet and a company's internal network.

DNS. Domain Name Services.

E2E. IBM's End-to-End System Design Method. See ISD.

ebAF. e-business Application Framework, formerly known as Network Computing Framework.

EBCDIC. Extended Binary Coded Data Interchange Code. EBCDIC is the 8-bit character encoding scheme used by IBM and compatible mainframes since the introduction of System/360 in the mid-1960s.

ECI. External Call Interface is a way to interface with CICS at a program to program level.

EJB. Enterprise JavaBean. An EJB is a non-visual, remote object designed to run on a server and be invoked by clients. An EJB can be built out of multiple, non-visual JavaBeans. EJBs are intended to live on one machine and be invoked remotely from another machine. They are platform-independent. Once a bean is written, it can be used on any client or server platform that supports Java.

EPI. External Presentation Interface is a way to interface with CICS at a program to 3270 level.

ERP. Enterprise Resource Planning

ESS. IBM's Enterprise Solutions Structure. ESS builds on the experience gained through E2E and enhances it through the concept of reusable standard solutions or "patterns" which are designed to be easily adaptable to new applications.

EXCI. External CICS Interface allows programs within MVS (including UNIX Services) to communicate with CICS programs.

Firewall. A computer - or programmable device - with associated software which can be used to restrict traffic passing through it according to defined rules. Controls would typically be applied based on the origin or destination address and the TCP/IP port number.

FTP. File Transfer Protocol

GUI. Graphical User Interface.

HTML. HyperText Markup Language

HTTP. HyperText Transport Protocol

IEC. The IBM International Education Centre in La Hulpe, Belgium.

ICM. Intellectual Capital Management. IBM's ICM AssetWeb System has about 50 knowledge network databases plus others such as the Issue-Based Structured Collaboration, several discussion forums, Project Profiles and Experiences, and so on, available for use by Global Services and Global Industries professionals.

IE. Microsoft's Internet Explorer.

IGS. IBM Global Services.

IIOP. Internet Inter-ORB Protocol

IMAP4. Internet Message Access Protocol Version 4.

IMS. Information Management System.

IP. Internet Protocol.

IPSec. IP Security Protocol. Provides cryptographic security services at the network layer.

ISAPI. Internet Server API.

ISC. Installation Support Centre, based in Hursley, UK. The EMEA ISC provides early education and support for new technologies.

ISD. InfraStructure Design. IBM's ISD is a path of the AD/SI method component of WSDDM. ISD is used to design the delivery platform supporting an application. ISD phases are Requirements, Architecture, Infrastructure Specification and Component Specification and Selection. ISD is based on the original E2E method which provided a formal way to design complex systems.

ISD. The Integrated Service Offering Development process of IBM Global Services.

ISP. Internet Server Provider.

I/T. Information Technology.

JavaBean. A JavaBean is a component that can be integrated into an application with other beans that were developed separately. This single application can be used stand-alone, within a browser and also as an ActiveX component. JavaBeans are intended to be local to a single process and they are often visible at runtime. This visual component may be, for example, a button, list box, graphic or chart.

JDBC. Trademark, often referred to as "Java DataBase Connectivity"

JDK. Java Developer's Kit.

JFC. Java Foundation Class.

JIT. Just In Time.

JVM. Java Virtual Machine.

LDAP. Lightweight Directory Access Protocol.

LEI. Lotus Enterprise Integration. This is the product formerly known as NotesPump.

LSX. Lotus Script Extensions.

LUW. Logical Unit of Work.

MIB. Management Information Base.

MIME. Multipurpose Internet Mail Extensions.

MOM. Message-Oriented Middleware.

MQ. Message Queue.

MVS. Multiple Virtual Storage. For many years the flagship operating system for IBM's large Enterprise Servers. In its latest releases formally known as OS/390, although the term MVS is still used unofficially.

NC. Network Computer or Network Computing.

NCF. Network Computing Framework.

NNTP. Network News Transfer Protocol.

NSAPI. Netscape Server API.

NSF. Notes database file extension.

NT. Windows NT (New Technology).

ODBC. Open DataBase Connectivity

OMG. Object Management Group.

OO. Object Oriented.

ORB. Object Request Broker

OS/390. The latest version of MVS. OS/390's standard OpenEdition function provides certified POSIX-compliant interfaces in addition to enhanced support for its traditional programming interfaces.

PERL. Practical Extraction & Reporting Language.

PGP. Pretty Good Privacy

PKI. Public Key Infrastructure.

POP3. Post Office Protocol 3

Port. A TCP/IP terminology, a port is a separately addressable point to which an application can connect. For example, by default HTTP uses port 80 and Secure HTTP (HTTPS) uses port 443.

RACF. Resource Access Control Facility.

RDMS. Relational Database Management System.

RFC. Request For Comment. Internet Standards are defined in documents known as RFCs.

RMI. Remote Method Invocation

RPC. Remote Procedure Call

RSA. Rivest-Shamir-Adleman algorithm.

SAP. Originally "Systemanalyse und Programmentwicklung" and now named Systems, Applications, and Products in Data Processing, SAP supplies widely-used software for integrated business solutions.

SES/workbench. A simulation product for behavioral and performance modeling of complex client/server, network, software and hardware systems.

SET. Secure Electronic Transaction.

SHTTP. Secure Hypertext Transfer Protocol.

SI. Systems Integration. One of the six competency areas of IBM Global Services.

SI/AD. Systems Integration/Application Development. One of the six competency areas announced by IBM Global Services in May 1998, later renamed simply SI.

SMTP. Simple Mail Transport Protocol

SNMP. Simple Network Management Protocol

SQL. Structured Query Language.

SSL. Secure Sockets layer

S/MIME. Secure MIME

TCP. Transmission Control Protocol.

TCP/IP. Transmission Control Protocol/Internet Protocol.

Telnet. U.S. Dept. of Defense virtual terminal protocol.

TME. Tivoli Management Environment.

TR. Token Ring.

UDB. Universal Database.

URL. Uniform Resource Locator. The standard way to address a location on the Web.

VA. Visual Age.

VAJ. Visual Age for Java.

VB. Visual Basic.

VM. Virtual Machine.

VPN. Virtual Private network.

VTAM. Virtual Telecommunications Access Method.

Widget. In this context, a generic term for something that can be put on a window such as a button, scrollbar, label, listbox, menu or checkbox.

WSDDM. IBM’s WorldWide Solution Design and Delivery Methods. WSDDM is a collection of generic plans and methods representing IBM's best practices for planning, managing, and delivering projects.

WYSIWYG. What You See is What You Get.

XML. Extensible Markup Language.

Index

O

P

R

S

T

V

W

ITSO Redbook Evaluation

An Approach to Designing e-business Solutions
SG24-5949-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and Fax it to: USA International Access Code + 1 914 432 8264 or:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
_ Customer _ Business Partner _ Solution Developer _ IBM employee
_ None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

_____

_____

_____

_____

SG24-5949-00
Printed in the U.S.A.

An Approach to Designing e-business Solutions

SG24-5949-00