

6

E-Commerce Systems Technology Infrastructure

Earlier chapters discussed vital components of e-commerce systems. This chapter introduces the readers to ancillary technology elements that an organization requires for the effective development and deployment of those systems. The distributed nature of e-commerce systems has further necessitated developments and innovations in technologies that bind the various components of these systems together as a functioning unit. Middleware is one of those components that offer ancillary services such as security, transaction processing, time services, and so on for the building of complex e-commerce systems. The design of e-commerce systems requires many middleware services, each of which fulfills a unique role for the effective functioning of an e-commerce system. This chapter elaborates upon the many services that fall into the realm of middleware. Directory services is another such component that enables systems and users to locate each other on the network. Various standards and products in this arena have challenged organizations to choose the right standard or product for their applications and business processes. This chapter presents an overview of the popular directory services commonly used in the development and deployment of e-commerce systems.

A groupware technology infrastructure is the cornerstone behind the establishment of virtual communities and the enabling of e-commerce applications with features that enhance communication, collaboration, and information sharing. Groupware includes features such as e-mail, information sharing over the Internet and intranets, and workflow. This chapter elaborates upon their role within e-commerce systems.

The multiple components and modules that shape an e-commerce system coupled with the dynamic business requirements necessitates component and object models for analysis and software development. The object-oriented stan-

dards and frameworks facilitate such development. This chapter delineates some of those models and development paradigms.

6.1 E-Commerce Systems Middleware

Middleware encompasses various technologies and products that facilitate the availability of backend network resources (e.g., databases) for frontend applications. Middleware components also include software that triggers backend applications to achieve end-to-end automation of business processes (e.g., chaining the order-entry business process with inventory and distribution processes). Middleware technologies facilitate integration with backend enterprise services that range from simple cases of information repositories to complex ERP applications (e.g., SAP R/3, Peoplesoft). An example in this case would be an e-commerce-based ordering system that interfaces with a production scheduling system that, in turn, ties-in to inventory, distribution, and shipping systems.

Multiple definitions exist for middleware. The IT industry has failed to standardize on the specific terms regarding middleware. The only definition that the industry tends to agree upon is that middleware is anything that resides between the client (user) and the server (database and application resources). These components could be data access components, communication protocols, specialized servers, or a mix of all of the above. The on-going evolution of computing architectures is one of the reasons for the ambiguous definition of middleware. Middleware played different roles in the three computing paradigms of legacy, client/server, and Internet-based computing. Transaction processing (TP) monitors were the primary middleware technology for legacy applications. The client/server era introduced multiple versions of middleware that included everything that resided between the two layers of the client and the server. Furthermore, a broader definition surfaced in the Internet era due to the vast distributed nature of Internet applications.

This book categorizes middleware as all the software components and applications that reside between the user (connecting through information appliance applications that solicit input from users) and the backend service repositories. These include access gateways, TP monitors, and other similar software. For example, if a user requests fulfillment of an e-commerce service through an information appliance, all software that resides between an information appliance (e.g., a PC) and the backend applications falls within the realm of middleware.

Middleware from the mainframe era primarily focused on the availability of backend resources. Middleware products such as transaction processing soft-

ware fit that description of middleware. The business application was a hodgepodge of business logic, network and database interfaces, and other application services. Only transaction processing software resided outside the realm of the business application to provide controlled access to databases.

The dawn of client/server computing triggered the emergence of additional middleware technologies. These include various database and network interfaces to link the various client and server pieces. This is where the definition of middleware started to blur. Distributed client/server computing resulted in additional middleware technologies and services.

E-commerce computing added more technologies and services to the middleware category. The primary reasons for this were the increase in the number of tiers that fall between the information appliances and the backend services and applications. Another reason was that the increase in the number of clients in e-commerce computing and the distribution of backend services across various networks required specialized network and application services. The popularity of building modular programs had decoupled these services from application programs and pushed them to the middleware pack.

With these trends, middleware has come to encompass various products, technologies, and services, some of which include:

- Access gateways;
- Database interfaces;
- Network and communication interfaces;
- Application interfaces to facilitate interoperability between distributed applications;
- Network/application services (e.g., security services, directory services, transaction services);
- Computer telephony integration (CTI) software;
- Middle-tier business logic implemented using traditional software technologies and programming languages or object frameworks such as Enterprise Java Beans (EJBs);
- Application execution services such as those that support large numbers of users, fault tolerance, workload balancing, session and state management, multithreading, accessing multiple resources, and others.

As mentioned earlier, middleware technologies and services play a vital role in e-commerce computing as they integrate a greater number of tiers in

order to materialize an e-commerce system. Middleware, for example, provides the following services to e-commerce systems:

- It supports diverse client-side environments such as Java clients, ActiveX clients, and other graphical environments.
- The decoupling of business logic from underlying application services hosted by different tiers requires new middleware technologies and services. Common Object Request Broker Architecture (CORBA) is an example of a framework that defines such services.
- By integrating applications and data distributed throughout the Internet, middleware enables a variety of IT-enabled services and products to customers. Organizations thus can offer better and more robust content by delivering an integrated view of applications.
- It supports various types of information appliances
- It integrates various tiers of e-commerce systems.
- It manages complex transaction flows.
- It can cope with the diversity of platforms (legacy, ERP, and others) and databases.
- It incorporates better failure detection and recovery capabilities.
- It handles an increased load of users.

Thus middleware technologies and products in an e-commerce framework enable the formation of business relationships through the integration of applications and services. The evaluation, selection, and acquisition of middleware is thus quite vital to build proper functioning e-commerce systems. The following describes the popular middleware frameworks.

6.1.1 Middleware Frameworks

Middleware frameworks are standards-based frameworks that enable the development of various middleware services. The two most popular camps defining these standards are Microsoft and the Object Management Group (OMG). The OMG defines various standards related to object and component technologies. Microsoft is pushing its COM architecture, whereas OMG is spearheading the CORBA initiative. This section reviews the two frameworks and their capabilities.

6.1.1.1 Common Object Request Broker Architecture (CORBA) Services

The OMG devised the CORBA specification, which is an object-oriented middleware specification facilitating object communication in a distributed paradigm. As described earlier, the middleware landscape includes numerous other services that include security services, naming services, and time services. CORBA not only defines the client/server middleware services present in the earlier generation of middleware products but it also provides a framework for the development of additional middleware services based on an object-oriented framework. The power of CORBA thus lies in enabling the building of distributed and object-oriented applications and facilitating interoperability between software objects regardless of their implementation details.

The object-oriented approach of the CORBA specification facilitates simpler integration of business applications. Organizations can focus on building objects that represent business functions and then combine these business functions with others through CORBA interfaces. This is unlike procedural middleware in which low-level APIs do not provide a higher-level abstraction and thus make the process of assembling applications more cumbersome.

OMG has defined the Object Management Architecture (OMA). OMA defines four primary middleware interfaces for the development of object-oriented applications: Object Request Broker (ORB), CORBA services, CORBA domains, and CORBA facilities.

ORB is what enables objects to communicate with each other. The ORB specification does not focus on the implementation of objects. Rather, it focuses on the interfaces that enable a client object to communicate with a server object. The interface definition language (IDL) represents the interfaces between objects. The Internet Inter-ORB Protocol (IIOP), on the other hand, is the communication protocol between ORBs. By invoking methods on another ORB through a client/server relationship, ORBs can communicate in a heterogeneous and distributed environment. This facilitates the development of objects regardless of implementation details such as platform, language, and location. When a client needs to communicate with another server object through an ORB, the ORB locates the necessary server object, establishes communication, invokes methods on the object, and returns the results to the client object that requested the services.

CORBA services are the system-level middleware services that are necessary to build enterprise-wide applications and systems. Table 6.1 presents some of the services defined by the OMG [1]. CORBA facilities provide higher-level services usable directly by application objects (e.g., printing). These facilities

Table 6.1
CORBA Services

CORBA Services	Short Description
Life Cycle Services	This service facilitates the life cycle services of objects, namely the creation, copying, deletion, and moving of objects.
Persistence Services	This service provides the interfaces that allow common operations to manage and retain the persistent state of objects.
Naming Services	This service enables the locating of objects, e.g., files, folders. It specifies the binding of names to objects and resolves those names.
Event Service	This service allows client and server objects to communicate. The event defines the supplier and consumer roles. The supplier object produces event-related data, whereas the consumer object processes the event data.
Concurrency Control Service	This service controls concurrent access to objects and supports both transaction and nontransactional modes of operations.
Transaction Service	This service supports transaction services as defined by the ACID properties of a transaction (defined in Section 6.1.2.).
Relationship Service	This service maintains the relationships between various objects, e.g., customer (object) buys (relationship) a car (object).
Externalization Service	This service is used for externalizing and internalizing objects.
Query Service	This service enables applications to query a set of objects and also allows for insertions, deletions, and other types of object manipulation.
Licensing Service	This service allows for the control of intellectual property rights on objects per business requirements.
Properties Service	This service allows for the manipulation of an object's properties.
Time Service	This service provides time-related services such as current time, interval between two events, order of events, etc.
Security Service	This service provides security features such as authentication, authorization, auditing, etc. to objects.
Trader Service	This service allows objects to interface with trader objects. Trader objects advertise an object's services to others (export) and also allow the matching of services on the request of a client object (import), so that the client object can use those services.
Collection Service	This service allows for the grouping of objects and enables a set of common operations on that group of objects.

include specifications for information management, systems management, and task management. OMG is working to define additional facilities related to mobile agents, data interchange, and internationalization. CORBA domains define specifications for specific vertical industries (e.g., finance, telecommunications, and manufacturing). These specifications define OMG-compliant interfaces for these vertical-industry systems.

CORBA does not limit itself to the definition of middleware services. CORBA also includes another component that defines the behavior of applications. This component is called CORBA Application Objects. CORBA Application Objects are not middleware services; rather they are the ultimate consumers of other CORBA services (i.e., application objects that are the users of middleware services). These objects are comparable to the Enterprise Java Beans (EJB) model described later in this chapter. Used in conjunction with EJBs, CORBA defines all object services necessary for the development of distributed object-oriented applications. CORBA defines these objects to enable the definition of business objects that are independent of the application details. These objects can define business constructs such as customer, order, payment, and so on. Various business objects can integrate and interoperate to define higher-level business processes (e.g., a stock trading application over the Internet).

Since the EJB framework defines the design of modular business objects within the Java paradigm, CORBA also defines the CORBA Beans framework to enable the integration of both technologies. The primary difference is that CORBA Beans framework allows the definition of business components in any language, as opposed to EJB, which relies on the Java language. CORBA bean containers can host and run EJB objects. Similarly, EJB containers can host and run CORBA bean objects if they are implemented in Java.

Several vendors have committed to support CORBA and have rolled out products that implement the CORBA specification. One of the first vendors to do so was IONA Technologies, which integrated CORBA support into its Orbix product. Orbix is a comprehensive implementation of CORBA that includes interfaces that facilitate the integration of software using CORBA middleware. IONA offers two Orbix products. The first, Orbix C++, is for C++ programmers, and OrbixWeb is for Java programmers. Orbix has found its way into the development and implementation of various types of vertical solutions such as telecoms, manufacturing, finance, healthcare, and many others [2].

Other popular products based on CORBA include IBM's Component Broker and BEA's WebLogic server.

6.1.1.2 Component Object Model (COM)

COM is Microsoft's object-oriented technology and infrastructure for the building of distributed object-oriented components and applications. COM grew out Microsoft's earlier Object Linking and Embedding (OLE) initiative. OLE provided a framework for building software components (called OLE controls) and the interoperability of those components. OLE's objective was to enable Windows applications to communicate effectively within the Windows environment by offering capabilities such as cut-and-paste operations and drag-and-drop functions.

The COM model provides various products and services for building distributed and object-oriented applications through COM products—unlike the CORBA architecture, which defines specifications for various services (e.g., naming and security services). Various products implement CORBA services based on these specifications. For example, the transaction service is an integral part of the CORBA services architecture/specification, whereas the Microsoft Transaction Service (MTS) product provides the same service in the COM architecture.

COM's design is closely tied to the Windows operating system, which is the primary reason for the popularity of the COM architecture. CORBA's design goal, on the other hand, was to be platform independent. However, Microsoft continually provides interfaces and gateways to open COM to other platforms. For example, COM has already been ported to various platforms such as AIX, Solaris, and OpenVMS.

COM+ is an extension of the COM model that further simplifies the creation of software components. COM+ adds runtime and services layers to the COM architecture. The runtime layer includes features such as dynamic invocation and memory management, and the services layer includes services such as security services, transaction services, and load balancing.

As both object-oriented frameworks (COM and CORBA) are quite popular in the industry, the OMG ratified a set of standards that enables interoperability between the two frameworks. Several tools are also available based on these standards that facilitate the integration of applications based on respective frameworks. OrbixCOMet is an example of an offering from IONA Technologies that is compliant with the OMG's standards and facilitates such integration.

6.1.2 Transaction Processing Middleware

Transaction processing programs facilitate controlled connectivity and service access to a large number of users with limited backend services. These include databases, other applications, and files. Transaction monitors link user applica-

tions executed on or through various information appliances to backend resources by providing services such as workload balancing, multithreading, and memory management.

Transaction monitors control the state of various transactions. Transactions are computer operations that change the state of network services such as databases. Their use in mission-critical applications is vital to ensure that changes to databases and other services defined within a transaction are either committed or left untouched. For example, a money transfer transaction involves deducting money from one account and adding it to another account. Transaction monitors ensure that both these operations defined in the transaction complete successfully (i.e., money leaves one account and goes to another account). If the transaction monitor detects a problem with any of these operations, it rolls back all operations and returns the databases and other modified services to their initial state.

TP monitors provide other services in addition to transactional integrity. TP monitors provide workload balancing, persistence, service management, queuing, fallback and recovery, and other services required for executing applications in a multiuser and multiservice environment.

The TP industry defines transactions as having four distinct properties, known as ACID properties. ACID stands for atomicity, consistency, isolation, and durability. Atomicity implies that all operations within a transaction form one unit of work. As explained earlier, the success of a transaction depends upon the success of all operations defined within the transaction. Failure of one requires the rolling back of all modifications. Consistency ensures that a transaction leaves the system in a stable state after its execution. If the transaction does not leave the state in a stable condition, all operations defined within the transaction abort and get rolled back. Isolation ensures that every transaction is unaffected by the actions of other transactions, while durability ensures that all operations of a transaction are permanent in nature.

TP monitors date back to the centralized mainframe applications era and have continuously evolved to meet the demands of new computing environments and architectures. With the emergence of the client/server computing paradigm, TP monitors bring the same transaction discipline to client/server applications. Popular TP monitors include BEA's Tuxedo, Transarc's Encina, and IBM's CICS.

With the emergence of e-commerce computing, TP monitors and associated technologies and standards evolved to handle e-commerce applications. For example, the new generation of TP monitors include support for ORBs. This enables TP monitors to bring ACID and other properties of TP monitors to the Internet and object world. In the object world, the discipline of transac-

tions becomes even more essential as invocation of one transaction may involve multiple databases and other network services spread across the Internet and enterprise intranets. Popular products from vendors include Microsoft's MTS, IBM's Component Broker, and BEA's M3. The new generation of TP monitors now support multiple backend resources, including relational databases, HTML repositories, and proprietary databases such as Lotus Notes.

Various technologies have surfaced to offer transactional support for object-oriented e-commerce applications. For example, as mentioned earlier, CORBA defines Object Transaction Service (OTS) as part of the CORBA services component. OTS enables distributed objects to access various transactional services. Java Transaction Service (JTS) is the Java implementation of OTS. JTS specifies the implementation of a TP monitor that supports the Java Transaction API at a higher level and implements OTS at a lower level [3]. EJB, to be discussed later, includes standards for the development of transactional beans. Microsoft, too, through its COM+ architecture, provides functions for the development of transaction processing applications on Windows platforms. All these object-based transaction processing standards facilitate the development of multitiered and object-based e-commerce applications.

The application development trends have continued to evolve with TP monitors as well. For example, TP monitor vendors have partnered with development tool vendors to facilitate the development of TP monitor applications from GUI tools. Similarly, various TP monitors have evolved to Web-enable their TP monitors, thus making them directly accessible to Internet and e-commerce frontends.

6.1.3 Communication Middleware

Communication middleware enables full-fledged applications to communicate with each other, unlike communication software that merely binds various modules of an application. The following delineates the popular types of communication middleware that facilitates such integration.

6.1.3.1 Message-Oriented Middleware (MOM)

Message-oriented middleware (MOM) is a class of middleware that enables client/server applications to communicate. MOM applications run by using queues. Each application places a message in the queue that the other application retrieves and then acts upon. Popular MOM products include IBM's MQSeries and BEA's MessageQ.

MOM applications differ from TP monitors as client and server applications do not maintain a logical connection. These applications are therefore

suited for time-insensitive applications. After the client application places a message in the queue, the server program can retrieve it at its leisure. MOM middleware allows programs to run offline, place messages in a queue, and then connect to the server for transmission of those messages. An example is a banking application that allows customers to open accounts. Since opening accounts requires banks to obtain certain authentication papers from their customers, accounts do not have to be available immediately for access on the Web. For such implementations, the bank can implement a MOM product that queues account-opening requests and later updates the bank's backend systems, thus alleviating the load of the backend systems.

Since MOM architecture suits different situations and applications, the OMG has introduced an object-based CORBA messaging standard using MOM that facilitates the development of object-based e-commerce applications and services.

6.1.3.2 Internet Inter-ORB Protocol (IIOP)

IIOP is a client/server protocol defined by the OMG as part of the CORBA specification for enabling CORBA objects to communicate over a network regardless of their location. The definition of the IIOP specification as part of the CORBA 2.0 specification facilitated the design of CORBA objects that can interoperate over the Internet. The IIOP protocol defines data formatting rules and various message types to enable ORBs defined per CORBA specifications to communicate with each other. The IIOP details are transparent to developers. IIOP works on various data transport protocols such as TCP/IP and SNA. IIOP facilitates communication between client and server object-based components (e.g., applets on the client-side and servlets on the server-side). IIOP thus provides an alternative to HTTP for facilitating application data flows between the client and server components.

6.1.3.3 Distributed Component Object Model (DCOM)

Similar to IIOP, DCOM is another protocol for interapplication communications across a network for communication with COM objects. DCOM is Microsoft's standard, which has in turn been licensed to other vendors. DCOM runs not only on Windows, but is available for UNIX, Macintosh, and other platforms. Developers can deploy software components across platforms distributed over a network and facilitate communication among the components using DCOM. DCOM is language-neutral and therefore a viable technology for building e-commerce systems that usually involve multiple application components developed in diverse languages on multiple platforms. These software components can include both applets and ActiveX components.

DCOM uses a DCE-RPC–based communication protocol to enable this communication and integration.

6.1.3.4 Hypertext Transfer Protocol (HTTP)

HTTP protocol is the Internet’s primary communication protocol that simply conveys information from the client (e.g., Web browser) to the server (e.g., Web server). HTTP primarily facilitates communication between an Internet’s Web browser and the Web server that hosts requests from the Web browser. HTTP is usually used to establish early connection between the browser and Web server. For example, Internet browsers use HTTP to request an HTML page from a Web server. Subsequently, application components such as JavaBeans, ActiveX controls, and applets—loaded through HTML files—communicate with backend server-side application components using other enhanced communication protocols such as DCOM and IIOP.

6.1.4 Database Middleware

Database middleware mask database-access complexities from applications, and also mask the implementation differences of various databases. For example, although Oracle and Informix are both RDBMS databases, they each provide native access mechanisms. Database middleware standards and products mask those differences from applications that access those databases and thus increase the portability of applications. Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) are both popular database middleware standards for the development of database access applications.

6.1.4.1 JDBC

JDBC, similar to ODBC, is an open specification that enables Java clients to access backend databases such as DB2, Oracle, and IMS. Using this method, Internet browsers, for example, can execute application components and directly connect to backend databases. Typically, JDBC is used to incorporate calls within backend application interfaces—such as servlets that access databases—and then send results to the client through access gateways such as Web servers. JDBC defines two interfaces. The first defines the application’s interface to a JDBC-compliant database, and the other defines the rules for database vendors to make their databases JDBC compliant.

6.1.4.2 ODBC

ODBC is an open database middleware standard developed by Microsoft that enables database access across various vendor databases. This means that any

client application can access an ODBC-compliant database as long as the client application uses appropriate ODBC drivers. Similar to JDBC, ODBC defines the application- and database-specific interfaces.

Major database vendors such as Oracle, Sybase, and IBM provide ODBC interfaces for their respective databases in addition to offering their own internal interfaces. For example, Oracle's native interface is called Oracle Call Level Interface (OCI), and Open Client is Sybase's native interface. Both also offer ODBC connectivity. Since ODBC is Microsoft's standard, Microsoft's SQL server uses ODBC as its native interface.

6.1.5 Application Middleware

Application middleware technologies and products enable the triggering of other applications, extend the functionality of applications, and provide various runtime execution services (e.g., memory management and workload balancing). The following delineates the various forms of application middleware used to develop e-commerce applications.

6.1.5.1 Common Gateway Interface (CGI)

CGI is the traditional way to invoke server-side programs, and it is portable. CGI programs are usually Perl scripts that invoke server-side programs developed in any language. CGI programs interface with the Web server using environment variables that the Web server populates by receiving input from the Web browser's URL. The Web server starts the CGI program that typically resides in a `cgi-bin` directory on the server platform. The CGI program interacts with the backend application or database, retrieves the result, formats it into an HTML or appropriate format and presents it to the Web server. The Web server then funnels the response to the browser for display. For example, a registration form filled out by a user while browsing on the Internet could be submitted to a CGI program on the server through the Web server. The CGI program, in turn, will process the form input by accessing a backend database and returning the results to the user's browser through the Web server.

CGIs are usually suitable for less intensive e-commerce applications. CGI programs are complete programs that invoke a separate process for each browser request and are thus inefficient for large-scale applications. This causes enormous overhead on the servers. CGI programs are not suited to run intense Web applications that service a large number of users and handle large transaction volumes. Nor are CGIs suitable for architecting and chaining complex applications that involve multiple transaction monitors and backend ERP applications.

6.1.5.2 FastCGI

Open Market, Inc. developed FastCGI to resolve some of the performance issues of CGI programs. FastCGI facilitates this by allowing programs to stay active in memory. FastCGI programs maintain persistent connections with applications and wait for server requests thus increasing the performance of applications. FastCGI compares favorably in performance with programs developed by the vendor's proprietary APIs (discussed next). However, FastCGI's primary advantage is its portability between servers, as most servers support CGI. Fast.Serv is a product from Fast Engines, Inc. that implements FastCGI for Netscape and Microsoft Web servers.

6.1.5.3 Vendor Proprietary APIs

Developers can use vendor proprietary APIs as alternatives to CGI scripts for invoking server-side programs. These APIs extend the functionality of various Web servers by allowing developers to write the additional modules needed for the execution of backend applications and programs. Developers can also develop modules that completely replace services that come bundled with various servers (e.g., Web servers). Most of the major vendors that sell Web-related servers provide APIs to extend the functionality of their servers. Popular examples include Netscape's Netscape Server API (NSAPI) and Microsoft's Internet Server API (ISAPI). NSAPI extends the functionality of all Netscape servers. Using NSAPI, developers can write modules or replace existing modules on those servers. Developers can write modules that provide various functions and services including security, logging, and other services.

Microsoft provides the ISAPI to extend the functionality of ISAPI enabled Web servers. For example, developers can use ISAPI to develop modules and applications that can retrieve data from backend databases. ISAPI applications are more efficient than CGI programs as ISAPI programs run in the same address space as the Web server and thus do not incur the overhead of communication between processes and invocation of processes.

6.1.5.4 Application Servers

Application servers embody various applications, network, and database services. Earlier application server implementations focused primarily on transaction-based services. However, the proliferation of Internet- and client/server-based services and technologies triggered the emergence of full-fledged application servers that encapsulate most functions in one middle tier.

Application servers provide capabilities for the development and deployment of server-side applications. These solutions are analogous to user interface development tools or integrated development environments (IDEs) that ini-

tially surfaced for the development of user interfaces. Application servers extend the same paradigm to networked and integrated applications. Some of the server-side application development and deployment issues—which would otherwise require extensive manual development and deployment practices—include the following features:

- Web server software functionality.
- Select groupware functions and services (e.g., support for e-mail, calendaring functions, newsgroups, and others).
- Transaction services functions.
- Runtime environment for the execution of various application logic, such as that of EJBs, COM servers, and others.
- Visual, IDE-like environment for the development of server-side applications.
- Interfaces for backend applications and diverse data sources.
- An integrated environment that enables the development, deployment, and integration of server-side applications.
- Ability to automatically partition applications based upon presentation, business, and data access logic.
- Application servers that handle browser requests and other requests from information appliances.
- Ability to facilitate the deployment of scalable applications across multiple servers and systems. Scalable deployment usually requires the moving of various system modules to various servers and the building of appropriate interfaces between various applications and modules of the systems. Application servers provide the basic foundation that allow for a flexible deployment application and system modules across servers and systems, yet provide a common glue that binds these modules transparently.
- Similar to providing an environment for scalability, a common platform base for application servers provides performance features such as automatic load balancing, stability of network and data connections, cooperative processing of multiple application components across platforms, and others.
- Numerous technologies exist for the integration of applications and ultimately the development and deployment of e-commerce systems. These include object technologies such as Java, CORBA, C++, IIOP, JDBC for data access, and so on. Application servers provide templates

and frameworks that facilitate a seamless development of applications that mask some of the complexities associated with the development of these applications and systems.

- Application deployment usually requires a security environment to control authentication, authorization, and other issues. Native application development requires the establishment of a security environment to provide such services. However, the application servers provide a security framework and environment for applications developed under the specific application server.
- Ability to integrate with legacy and ERP applications such as PeopleSoft and SAP.
- Support for multiple databases.

6.2 Directory Services: Glue for E-Commerce Systems

Directory services store information about network resources and the users that access those network resources. No sooner did enterprises start linking computers on networks that the need for directory services emerged. Users needed a means to locate other users to transmit information and needed to locate other computers to access the various services resident on those nodes. Enterprises implement directory services to connect various users and applications. Following are some of the typical applications that require directory services:

- An e-mail directory enables users to locate other users for sending e-mail. Today this directory exists in many forms. Enterprises often maintain a unified e-mail directory for proprietary deployed e-mail systems, or in the case of Internet e-mail addresses, users maintain local address books on their information appliances. Similarly, public portals provide a set of services for locating Internet e-mail addresses, which either locate e-mail addresses present on their domains or attempt to provide a listing of e-mail addresses present in the Internet universe.
- A LAN directory stores user information about people on the LANs and facilitates various functions such as connecting to the Internet, sharing printers, facilitating LAN chats, browsing through information published on the LAN and the intranet, and so on. Common implementations include NT domain directory infrastructure, Netware Directory Services (NDS), and so on.

- Developers require access to source code repositories to check out source code. Various source code configuration management tools maintain various information about developers in a directory to enable them access to such intellectual assets.
- To access networks through firewalls, enterprises require users to authenticate to the firewall database that stores information about the users and their respective access rights.

A directory services platform includes various components such as:

- *Lookup database:* This is the database that stores information about network resources and user profiles. A user profile includes a user name, address, department code, telephone number, and so on. Network resources and services information includes printer names, computers, applications, and others.
- *Lookup functionality:* These functions enable the requesting of programs and services to get information about network resources and services.
- *Security services:* These services ensure that all accesses are within the defined security limits as set and defined by the security administrator. A directory service can employ various forms of security controls that can range from simple static password authentication to public-key cryptographic systems. Chapter 8 discusses various forms of authentication schemes.

E-commerce applications further fueled the need for additional directories. E-commerce computing relies on the existence of one large ubiquitous computing network with users and network resources. Carrying out transactions and various services as well as locating and searching for information on this large network demands the existence and establishment of various directory services at various legs of the network. For example, the following describes sample scenarios within e-commerce computing to demonstrate the importance of directory services:

- The establishment of a secure application infrastructure based on public-key cryptography requires the existence of a public-key directory that includes the names and certificates of relevant users. For example, connecting a browser to a server using an SSL session involves authentication of the key loaded onto the browser. This

authentication is performed by the certificate authority that maintains certificates for the key loaded on the browser.

- An organization sets up an extranet for external partners and suppliers to transact business-to-business e-commerce services. A directory service can be used to store the appropriate security credentials of all parties that have access to the extranet resources. The directory service handles all login requests and provides users with access to appropriate network functions.
- Users on the enterprise's intranet that require access to various network services (access to applications located on various network servers) and devices. For example, printers can use a directory service that facilitates the means to locate those resources on the network and enables transparent connectivity.
- To access corporate mission-critical applications, users can sign-on to a portal or portal-like directory that provides users with a view of authorized applications.
- Various e-commerce applications may require their respective directory services. For example, an organization that provides Internet bill payment functions may use a directory that enables customers to sign-onto the to Web site to make appropriate payments.

Enterprises usually maintain multiple directory services for different applications, which has resulted in islands of directories. This trend is on the rise, as enterprises aggressively launch various e-commerce services that require the deployment of relevant directory services. Almost all e-commerce transactions and services require the establishment of directory services.

The problem of multiple directory services and the inability to effectively locate network resources through directory services will continue to worsen unless enterprises launch a formal initiative to resolve these issues. The directory service problem has worsened over the years due to the numerous applications and associated computing platforms that have surfaced. It is therefore vital that enterprises formally decide upon the particular strategies for directory services deployment. Some of the issues associated with multiple directory services are as follows:

- Users must sign-on multiple times to use various network services.
- Users cannot logon due to lack of streamlined administrative processes that cover multiple directories. For example, a user may be allowed to

use the e-mail services in a directory but is denied access other enterprise applications.

- Users cannot locate appropriate services or applications.
- Users find it difficult to launch enterprise applications because they need to interface with multiple directories that require multiple levels of authentication.
- Organizations have to implement multiple security administration processes to enable synchronization of those directory services spread throughout the enterprise.

6.2.1 Centralization of Directory Services Through Metadirectories

The preceding discussion is indicative of the dilemma that enterprises face in moving to an e-commerce computing paradigm, which requires a more unified way to locate network resources. An obvious option is to centralize directory services. The advantages of deploying a unified directory services infrastructure are many. Based on the issues that enterprises have with the islands of directory services, one can clearly deduce the advantages of such a solution. Centralized directory services provide users with a single sign-on (SSO) solution, because a one-stop authentication service provides users with access to all authorized functions. A centralized directory also simplifies management of various functions (e.g., definition, modification, and deletion of users and network resources).

Whether centralized directory services would result in the appropriate ROI or not depends upon various factors, which include:

- Nature of the enterprise's business processes (more applications usually result in more directory services).
- Forecasted business growth (increasing number of users requires scalable directory services).
- Challenges that enterprises are facing with the current directory service infrastructure.
- Nature of e-commerce applications planned for the future.
- Various platforms installed in the enterprise may drive the choice of a strategic directory services platform, as certain directory service frameworks may be tied to specific platforms.

A popular option for centralizing directory services is to implement meta-directory services. Metadirectories logically or physically unify various directory services and their contents that include people, roles, their credentials, department codes, and so on. Metadirectories, besides providing a unified information model for the representation of namespaces, also reduce maintenance costs. For example, the addition or deletion of an enterprise employee could require the updating of multiple directory servers (e.g., the human resources directory, e-mail directory, and enterprise application directory servers). Metadirectories achieve this integration by maintaining relationships between various directories. An enterprise can therefore unify its directory services that may include NDS, NT directory services, IBM Resource Access Control Facility (RACF), Lotus Domino, Netscape's directory services, and many others.

An enterprise can implement the concept of metadirectories in various ways. One method involves acquiring (or building) a product that automatically synchronizes directory information with multiple directory services. For example, this product would automatically update the information of a new user in multiple directory services, alleviating the requirement to create the user in multiple directory services. Another method involves a procedural approach in which directory service entries in one server replicate to multiple directory services as a batch process, thus making basic user information available in multiple directory services.

Isocor, a leading provider of metadirectory solutions, offers a product called MetaConnect. This product provides metadirectory functionality by reducing the maintenance associated with multiple directory services. The product is based on the Lightweight Directory Access Protocol (LDAP), and operates with most LDAP servers, thus centralizing most directory-services-related functions (e.g., user addition, deletion, profile management). MetaConnect's architecture enables linking to disparate directory servers and centralizing information into one model (e.g., map the "user address" and "address" fields in multiple directories to one logical attribute). This provides a unified view of various directory services. MetaConnect unifies directories by storing all user information from multiple directory servers into one metadirectory (e.g., e-mail address in one directory and telephone numbers in another directory). These features along with others alleviate various management and administrative burdens associated with the maintenance of multiple directory services.

6.2.2 Directory Services Requirements for E-Commerce Systems

As mentioned earlier, the primary functions of a directory service are to enable the definition and lookup of users and network resources. E-commerce computing requires various features from a directory service that provide e-commerce services to its users, including the following:

- *Scalability:* Certain e-commerce implementations, especially in business-to-consumer domains, require the definition of a large number of users in order to give them access to appropriate e-commerce services. For example, an organization providing online bill-payment services will need to define all of its users' profiles in a centralized directory service. This directory should be scalable to meet the exponential increase in customers. Scalability should therefore be a factor in the choice and deployment of appropriate directory services.
- *LDAP support:* As will be explained later, LDAP has become a directory access standard in the world of directory services. The directory service should therefore support the LDAP standard to allow interoperability between various directory systems on the Internet and intranet.
- *Architecture robustness:* The directory service should offer a robust architecture to enable a richer definition of users and network resources. For example, the definition of a printer should allow for its various properties, physical location, and so on, such that a user looking for a color printer should be able to locate the appropriate one. The directory service should also represent relationships between various entities to facilitate lookup of entities and their respective properties. Furthermore, a directory service's architecture should be flexible enough to represent organizational structures. For example, the directory service should require minimal actions to redefine a user in another department with a different set of credentials. NT 4.0's directory model has proven to be complex enough to cater to such features in large environments.
- *Performance:* The directory service should perform adequately under a large number of network connections. Performance becomes an even bigger issue if the server performs complex operations before fulfilling a user's request. For example, a server could suffer performance degra-

dation if it performs complex security functions (e.g., SSL) for every transaction. To handle such scenarios, certain directory vendors provide interoperability with hardware accelerator boards on the servers that boost the performance of such directory servers.

- *Fault tolerance:* Directory services constitute a vital link for accessing critical e-commerce services. The directory service therefore should have built-in fault tolerance or the organization should build alternate measures of fault tolerance to avoid business unavailability problems for its Internet-based customers.
- *Management issues:* Management and administration are vital issues in deploying directory services. User and object additions, deletions, and modifications of profiles can add up to an enormous number of tasks. The administration of a directory service solution for a given application or business process should not incur high costs. For example, NT directory service builds on a domain and trust relationship model. This model works quite well for small numbers of users and domains, but the administrative burden to manage an enterprise with a large number of users and domains becomes extremely burdensome.

6.2.3 Popular Directory Services

The past few years have witnessed the emergence of numerous directory services standards. This section reviews some of the popular standards and products related to directory services.

6.2.3.1 X.500

X.500 was one of the early initiatives that surfaced to standardize various directory services. The X.500 standard is the culmination of the efforts of ITU and ISO in the early 1980s. The X.500 directory service standard surfaced to address the complex operations inherent in the design of a directory service. However, due to the complexity and bulkiness of the protocol, most vendors did not fully implement the X.500 protocol, resulting in incompatibilities among various implementations and a fragmented market of directory services. Vendors that followed the standard in various forms include Microsoft (Exchange), Lotus (Notes), and Novell (Netware Directory Service).

6.2.3.2 Lightweight Directory Access Protocol (LDAP)

Netscape along with some university researchers defined the LDAP standard in mid-1990s. LDAP is a protocol for accessing X.500 directory services and runs on TCP/IP. RFC 1777 defines the initial specification for LDAP, while RFC

2251 through RFC 2256 define specifications for LDAP v.3. The primary reason for the definition of LDAP was to define a lightweight version of the X.500 standard, as the Directory Access Protocol (DAP) implementation of the X.500 standard, due to its complexity and bulkiness was challenging to use and implement. For example, LDAP queries generate much less traffic than the X.500 DAP. The DAP component of the X.500 defines the access mechanism between the client and the server. A server that supports LDAP is the Directory System Agent (DSA). The term *DSA* is also based on the X.500 terminology. The DSA submits the user's query to multiple directories and retrieves the response from the LDAP-compliant directories. The LDAP standard thus facilitates the integration of disparate directory services such as NDS, NT, and so on, and is a lightweight implementation of DAP that enables lightweight clients to access X.500-based directory services over the Internet.

LDAP's directory schema uses a tree-like structure to represent its entries. This tree is also called the directory information tree (DIT). The topmost entry is the "root" that includes subhierarchies of countries, organizations, people, and so on. The entries in an LDAP server can vary depending upon the specific implementation. For example, an extranet application can implement an LDAP server to authenticate its users. The LDAP supports basic commands that can be executed against an LDAP-compliant server. These commands enable search of the directory for objects, maintenance of objects in the directory (add, modify, delete, etc.), establishing sessions with an LDAP server for communication, and many more. These commands can be used between client applications and LDAP servers as well as between various LDAP servers that collectively implement a larger directory of information.

LDAP version 3 adds new features to the original LDAP protocol. These include better security features and a robust command set. The security features include the addition of a Simple Authentication and Security Layer (SASL) for additional authentication. The enhanced command set enable LDAP clients to discover the directory schema of the directory database, unlike the earlier version of LDAP in which clients had to have the knowledge of the directory schema before accessing the directory entries.

LDAP has become a very popular standard, and most directory vendors support it. These include Novell's Netware Directory Server (NDS), Microsoft's Active Directory Server (ADS), and Sun-Netscape's iPlanet Directory Server.

6.2.3.3 NDS 8 Directory Services—Full Service Directory (FSD)

Novell has been an active technology supplier of enterprise directory services for a number of years. The vendor enjoys a large customer base when compared

to other players such as Microsoft. Novell's popular directory services architecture is called Netware Directory Services (NDS), and it forms the backbone of numerous large enterprises. NDS emerged as a popular directory service primarily due to the large installed base of Netware networking software.

Novell recently introduced NDS 8 and refers to it as a full-service directory (FSD). Novell's goal in introducing this directory service was to incorporate all the desired functions of an ideal directory service. Although this approach seems quite promising, the biggest challenge in introducing directory services in an enterprise is its interoperability with other directory services.

Novell clearly defines four issues as being vital in a directory service and defines appropriate functions within those areas [4]. Those areas and appropriate functions are as follows:

- *Discovery*: Discovery is the process of lookup of network resources. A user or application should have the ability to appropriately browse and search the various contents of a directory service. NDS 8 provides appropriate features for the discovery functions and employs the LDAP standard for this function.
- *Security*: NDS 8 defines robust levels of security. A directory service should provide appropriate security services for the information stored in the directory service and for the storage of security information to enable users to engage in various security functions such as secure e-commerce transactions.
- *Storage*: A directory's database structure is equally important as it not only stores information about users and network resources but provides the appropriate structure for the storage of this information.
- *Relationship*: An essential feature of any directory service is its ability to store relationship between various entities. NDS 8 provides a robust infrastructure for the definition of such relationships.

NDS 8 provides the following distinctive features:

- The ability to define a large number of objects (e.g., users and other network resources). NDS claims this number to be close to a billion objects. This provides organizations with the benefit of deploying NDS in the business-to-consumer paradigm, which may require the definition of a vast number of users that access an organization's services.
- From a performance viewpoint, NDS 8's architecture measures adequately for this number of objects.

- NDS 8 builds upon the latest version of LDAP (version 3).
- NDS 8 provides robust tools to import a large number of directory entries into its core.
- NDS has the ability to run on the NT system and override NT's directory service by providing NDS services. This is especially useful for environments that have mixed NDS and NT environments and plan to migrate toward a full NDS implementation.
- NDS provides administration and management tools that meet the challenges associated with the definition of a large number of network objects.
- NDS supports the JNDI initiative.
- NDS provides transparent replication services that allow users to access profile replicas from various locations. This enables an enterprise to provide adequate support for users who access network resources through various mobile information appliances from various locations.

6.2.3.4 Microsoft's Active Directory Services (ADS)

Microsoft's latest initiative in the area of directory services is the Active Directory initiative. Active Directory is an integral part of the Windows 2000 initiative and will replace the earlier NT platform directory services and structure. Microsoft introduced the Active Directory initiative to resolve various issues inherent in the NT directory services and to cater to the new directory services required in the e-commerce computing paradigm. Specifically, Active Directory provides the following primary features [5]:

- Facilitates the establishment of a virtual network by providing a unified view of the network.
- Compared to NT directory services, ADS requires a reduced number of trust relationships for managing domains.
- Provides a mechanism to enable the establishment of a unified messaging structure. This enables the definition of attributes related to various messaging applications (e.g., e-mail, telephone, fax, and others).
- Provides a mechanism for managing other directory services without regard to their platform details.
- Provides an API for accessing other directories.
- Provides support for open standards.
- Builds upon the LDAP protocol for accessing directory services.

- Provides the Active Directory Services Interface (ADSI) specification to enable external applications to access the Active Directory service.
- Active Directory, like most other directory services, does not conform to the X.500 standard but does incorporate certain features.
- Active Directory incorporates Microsoft's Intellimirror management technology, which provides administrators with various system management features. These include automatic software distribution and maintenance, centralized desktop configuration management, and remote operating system installation.
- Active Directory is backward-compatible with NT directory services.
- Provides various security features that include kerberos authentication, x.509 certificates, and LDAP over SSL.

The debut of Windows 2000 presents users with an opportunity to standardize their LAN-based directories, especially if they are based on NDS and previous versions of NT directory services. Microsoft and other vendors are offering numerous tools that enable organizations to achieve synchronization between various directory services and ADS as well as to migrate from diverse directory services to the ADS. For example, Microsoft Directory Synchronization Services (MSDSS) synchronizes data between NDS and ADS. However, since directory services are a key infrastructure element of an organization's network strategy, organizations should wait for the maturity of ADS and test directory migration strategies before rushing into a mass migration of users and network resources and standardize on ADS. This prudent course of action for migration should be followed not merely for non-Microsoft-based directory services such as NDS but also applied to migration from NT directory services to ADS. The reason is that although ADS is backward-compatible with NT directory services, numerous new features and new security modes in ADS necessitate detailed planning.

6.3 Internet Domain Name Service (DNS)

DNS is the Internet's directory service. DNS facilitates the unique identification of an organization or entity on the Internet. DNS maps the domain name of an organization (e.g., `www.shopname.com`) to its IP address (e.g., `123.456.789.012`). Various DNS servers distributed across the Internet facilitate the mapping of the friendly domain names that users enter into their browsers to connect to an IP address. An organization may install local servers that connect to the Internet name servers to resolve a domain name. On locat-

ing the IP address, the Internet application then uses that IP address to route the information to the appropriate hosts.

DNS assignment tasks include:

- *DNS name registration*: This refers to obtaining “friendly” names for e-commerce sites (e.g., yahoo.com). Network Solutions, Inc. (NSI) has primarily acted as the default global registrar for the .com, .edu, and .net domains. Enterprises obtain domain names from NSI. Recently, however, other registrars (e.g., register.com) have begun to provide comparable services.
- *IP address block allocation*: Internet connectivity requires every host to have a unique IP address on the Internet. This activity deals with maintaining the entire IP address space of the Internet and with allocating of unique IP addresses to Internet hosts.
- *Protocol parameters*: This activity deals with the assignment of miscellaneous protocol parameters and values required for the seamless operation of the Internet.

DNS maintenance comprises multiple activities. Although the primary control of DNS has been under U.S. government’s supervision, various organizations have also provided these services. Recently, the U.S. government has triggered initiatives that promise to push DNS and its activities into the public sector. This section provides the background and current state of affairs of the DNS control.

6.3.1 DNS Structure

DNS is a hierarchical database that consists of various domain names (e.g., yahoo.com, artechhouse.com), their respective IP addresses, and other relevant information. DNS uses a tree-like structure for organizing data fields. This is similar to the UNIX or DOS file system. For example, in the DOS file system’s directory listing C:\ABC\DEF, “C” is the top-level name (drive name), and ABC is the directory name under that drive. DEF then falls under the ABC directory. The DNS follows a similar convention but represents the hierarchy in the reverse order. For example, for the domain name yahoo.com, “com” is the top-level domain (TLD), that includes yahoo as the subdomain or second-level domain (SLD). TLDs are of two types: generic TLDs and country-code TLDs (ccTLDs). Table 6.2 describes the generic TLDs and their respective meanings. The country code domain names were created for use by entities

Table 6.2
Generic TLDs

TLD	Description
com	Commercial entities (e.g., www.amazon.com)
org	Nonprofit organizations (e.g., www.icann.org)
net	Networking organizations (e.g., www.gte.net)
edu	Educational institutions (e.g., www.mit.edu)
gov	Governmental institutions (e.g., www.irs.gov)
mil	Military institutions (e.g., www.darpa.mil)
int	International treaty institutions (e.g., www.reliefweb.int)

within each country. Examples of country code domains are .uk (United Kingdom), .jp (Japan), .fr (France), and .dk (Denmark).

The Internet includes millions of hosts, and access to any of these hosts requires DNS entries to remain up-to-date. Thirteen DNS root servers distributed around the world provide transparent access to any host on the Internet. Local DNS servers subscribe to these servers and obtain the relevant information to connect to Internet hosts. The U.S. government controls these root servers. The thirteen root servers are A.root-servers.net, B.root-servers.net, and continue through M.root-servers.net. These servers are distributed throughout the United States and Europe, although most of them are located in the United States.

The DNS database (also referred to as the registry) consists of resource records (RR) to handle the mapping of IP addresses and host names and other information. Each RR within the DNS registry includes the following elements:

- *Type*: Indicates the type of resource. Numerous values can appear in this field. For example, if the RR contains the value MX (Mail eXchanger), it would indicate the name of the mail servers that receive Internet mail on the host's behalf. Similarly, if the type is A, it designates the IP address (or addresses) of that host name. Other types include HINFO (Host information), CNAME (Canonical name or nicknames or aliases), and TXT (Text strings), which can include any type of information.
- *Class*: The protocol family of this record. For all Internet records, the indicator *IN* is used.

- *TTL*: Time to live indicators hold time intervals that various name servers use to cache data and to poll other servers to get updates
- *Data*: This represents the data in the field and depends upon the Type and Class fields, as different values of Type and Class will have data pertaining to those fields. For example, for the MX type, the data would indicate mail server names, and for the A type, the IP address of the host would be included.

6.3.2 History of DNS Operations and Organization Control

Chapter 3 explained the networking structure of the Internet, which highlighted the Internet's structure as a conglomerate of independent networks that interconnect through entities called NAPs and ISPs. The existence and overseeing of NAPs is essential to ensure the seamless functioning of the Internet. Initially, the U.S. government had full control of the NAP architecture. However, most functions have shifted toward industry in general and are controlled by ISPs. Similarly, DNS was initially under the full jurisdiction of the U.S. government. However, the government has recently launched initiatives to privatize DNS operations to enable competition and promote innovation in the DNS system.

Before delving any further into the current state of DNS-related activities, it is vital to review the background information and history of DNS. ARPANET developed the DNS system in 1985 to cope with the exponential growth of Internet users and network activity. The formal specification for the DNS (RFC 1034) was published by Jon Postel at the University of Southern California's (USC) Information Sciences Institute (ISI) in 1987. This specification provided detailed standards for the technical design of the domain system and other related issues. Postel also maintained the addresses of the Internet working in conjunction with SRI International, which was the primary contractor to the U.S. government for this activity. With increased activities, these functions resulted in the formation of the Internet Assignment Number Authority (IANA).

The U.S. government later delegated some of the DNS-related services to other organizations. For example, NSI (formerly known as InterNIC) started to handle the domain name registration process. In 1993, the U.S. government officially began funding NSI. Later, in 1995, NSI started charging for the DNS services and became the sole entity to profit from DNS services.

Until very recently, NSI was responsible for most tasks associated with DNS administration, including registering domain names, updating the registry (DNS database), performing registrar services (domain name assignment),

and controlling the root name servers. NSI performed these activities under a cooperative agreement with the U.S. government. NSI has been the primary entity to provide domain name registration services for the .com, .net, and .org TLDs and ccTLDs. NSI maintains the master file of domain names and distributes the domain data to the thirteen root servers around the world.

NSI also controlled the task of allocating IP addresses. The U.S. government funded the operation of NSI for this endeavor. IANA was the primary governmental body that authorized the NSI to allocate IP address spaces. Later, yielding to pressure from the Internet community that its members should control the address space allocations, NSI helped form separate bodies by getting appropriate approvals from the National Science Foundation (NSF). These bodies were under IANA's control, and ISPs contacted these bodies for IP allocation services. IANA allocated larger IP address blocks to regional Internet registries (RIRs), which in turn allocated IP addresses to large ISPs. ISPs suballocate those addresses to smaller ISPs, which in turn suballocate specific addresses to end user organizations [6].

The following bodies control IP address allocation:

- American Registry for Internet Numbers (ARIN), in the United States, South America, and other regions (www.arin.net);
- Reseaux IP Europeans (RIPE), in Europe (www.ripe.net);
- Asia Pacific Network Information Center (APNIC), in the Asia Pacific region (www.apnic.net).

NSI still supports the ARIN initiative. However, once ARIN becomes fully independent, NSI will relinquish all control. This means that all organizations that want an IP address on the Internet (within the IPv4 address space) will have to contact ARIN in the United States. However, in reality only larger ISPs directly request these services from ARIN. These larger ISPs then allocate IP addresses (within the authorized address space) to smaller ISPs, which then allocate address spaces (all within the limits of the larger address space allocated by ARIN) to other smaller businesses and customers.

6.3.3 Future Directions for DNS Operations

In 1996, the ISOC (Internet Society), IANA, and other governmental and public sector entities formed the International Ad Hoc Committee (IAHC), to launch a process for the self-governance of the Internet and competition after the NSI monopoly ended. The IAHC later evolved into the Policy Oversight

Table 6.3
New Generic TLDs

TLD	Description
.firm	Businesses or firms
.shop	Businesses offering goods to purchase
.web	Entities emphasizing activities related to the Web
.arts	Entities emphasizing cultural and entertainment activities
.rec	Entities emphasizing recreation/entertainment activities
.info	Entities providing information services
.nom	Entities wishing individual or personal nomenclature, i.e., a personal <i>nom de plume</i>

Committee (POC), with a Policy Advisory Board (PAB) to lead the evolution to self-governance.

The process resulted in a generic Top Level Domain (gTLD) Memorandum of Understanding (MoU). More than 200 organizations throughout the world signed the MoU. The MoU led to the creation of the Internet Council of Registrars (CORE), a nonprofit organization for administering seven new TLDs identified during the MoU process. Table 6.3 describes the additional TLDs.

The U.S. Department of Commerce, through a white paper published in June 1998, requested the establishment of a nonprofit organization to perform administrative activities related to domain names and management of the Internet's address space. Later, in November 1998, the Department of Commerce recognized the Internet Corporation for Assigned Names and Numbers (ICANN) as that organization by signing a Memorandum of Understanding [7]. ICANN has taken over the U.S. government's role and function (earlier performed by IANA) for such functions. IANA has therefore moved under the ICANN umbrella.

ICANN defines itself as the new nonprofit corporation formed to take over responsibility for IP address space allocation, protocol parameter assignment, domain name system management, and root server system management functions now performed under U.S. Government contract by IANA and other entities [8].

The U.S. government white paper distinguishes the term *registry* from *registrar*. In essence, the registry is the database that contains DNS entries, whereas the registrar accesses the registry for various activities (e.g., defining a

new second level domain (SLD) name under the .com, .net, and .org TLDs). ICANN's primary objectives are the formulation of policies and procedures for the establishment of registries and registrars to manage and control the root name servers. ICANN is responsible for the accreditation of the registrars that register the Internet domain names. Since registering domain names is a critical activity for e-commerce sites, ICANN is formulating various guidelines for the qualification process. ICANN has also proposed itself to be operationally involved in the process for valid registrars. If the systems providing those services fail or if the DNS registrar fails, ICANN will provide appropriate resiliency and minimize the impact on e-commerce sites and businesses.

One of the objectives of ICANN is to open competition for the registering of domain names. ICANN has focused on taking a pragmatic approach to the problem, as decentralizing the process is prone to surface many issues. For example, domain name holders require the assurance that their domain names will resolve to the appropriate IP address anywhere on the Internet and will be stable. Traditionally, since only one entity (NSI) managed the process, there were lesser issues.

As mentioned earlier, NSI has always been the default registrar. ICANN is currently pursuing initiatives to open the domain name registration process to other registrars as well. ICANN is responsible for formulating guidelines for the accreditation of various registrars and to accredit registrars based upon those guidelines. To ensure a smooth transition, ICANN initially selected five registrars that will operate and sell domain name services alongside NSI. In parallel, the ICANN has also begun to accredit additional registrars that will take over operations once the transition phase is complete. NSI's responsibility in this transitional opening of the DNS to competition is to relinquish control of its registry by creating a Shared Registration System (SRS) that will enable new registrars to access the DNS registry in the same manner as the NSI itself. As of this writing, Register.com (www.register.com) has become the first registrar to offer registration services per the new guidelines defined by ICANN. Other registrars to take part during the transition phase include America Online, CORE (Internet Council of Registrars), France Telecom/Oléane, and Melbourne IT [9].

Other registrars will also provide generic TLDs to define domain names other than the .com, .net, and .org TLDs. For example, CORE, another registrar selected by ICANN, will provide the names .firm, .shop, .web, .arts, .rec, .info, and .nom, which will be available through independent CORE registrars on five continents [10].

6.3.3.1 ICANN Structure

To address the three issues of domain name registration, address space allocation, and protocol parameters, ICANN has formed three supporting organizations to help address policy matters and other issues. The three organizations to address those issues are the Domain Name Supporting Organization (DNSO), Address Supporting Organization (ASO), and Protocol Supporting Organization (PSO). DNSO advises ICANN on issues related to the domain names used for referencing Internet sites and deals with the appropriate policy issues. ASO ensures the uniqueness of the assigned addresses. Internet IP addresses are similar to telephone numbers and provide connectivity to the Internet. The current Internet address assignment is controlled by the three regional internet registries (RIRs) mentioned earlier (ARIN, RIPE, and APNIC). ASO will perform most of the functions of the RIRs and may formulate additional policies per ICANN directives. Under the new model, ICANN will allocate the address space and delegate the responsibility to further subdivide the address space to RIRs [11]. PSO deals with general issues related to communication over the Internet.

As of this writing, most of the aforementioned activities are still in transition. NSI still manages most of the DNS related activities and is in the process of relinquishing control, thus enabling other organizations to actively participate in the IP allocations and domain name addressing functions. Deregulation of these activities and opening the process to competition will simplify the various steps involved in registering e-commerce sites and will provide organizations with more robust services from the new registrars.

6.4 Enabling Groupware for E-Commerce and the Internet

The term *groupware* encompasses many things. Groupware can be considered a group of technologies that enables various members of the value chain of an organization to communicate, share information, and perform activities per defined workflows. Communication in this context can take many forms, including the sending of e-mail and interactive conferencing through various information appliances. Information sharing refers to accessing archives of historical discussions and e-mail, corporate policies, and so on, as well as sharing information through whiteboards and other collaboration tools. Workflow refers to defining work steps through the implementation of various business rules (e.g., filling purchase requests, submitting them for approvals to other parties, routing them to the next step, etc.).

Historically, groupware tools and technologies have been considered part of the communications category. However, recent advances in groupware tools incorporate functions from other categories as well. Still, no groupware vendor packs all functions that fit the above-mentioned three categories into one package. Various solutions from numerous vendors have surfaced that seem to offer different pieces of the groupware puzzle.

Groupware tools and infrastructure facilitate the automation and streamlining of an organization's business processes not implemented in an organization's legacy and ERP systems. An organization can use groupware tools to streamline and automate intrabusiness processes that may not be ideal candidates for implementation through larger-scale application frameworks (e.g., ERP systems and applications). For example, an organization can implement a groupware-based procurement application that fulfills an organization's basic procurement needs. The groupware application can provide the basic workflow (incorporating rules for approvals) and basic communication tools (e-mail for sending requests, sending alerts on approvals) for implementing the application. Conversely, an organization may implement a large-scale EDI/e-commerce application to automate and streamline the organization's procurement process, which involves millions of transactions. Groupware applications thus complement enterprise applications implemented on ERP and other legacy platforms. Enterprise applications automate business processes for large-scale and high-volume data business applications. Groupware products, on the other hand, provide workflow functionality and automate business processes that rely on functions such as electronic communication, meetings, discussions, document forwarding, and so on.

Groupware has undergone three phases of evolution. In the first phase, the term *groupware* did not exist. However, functions that make up groupware today existed as separate applications. For example, e-mail, calendaring software, and other similar applications existed as separate applications. The second phase gave birth to the term groupware, and it encompassed all applications that enabled teams to communicate, work, and share information. Groupware packages provided organizations with the opportunity to structure activities that relied upon the usage of those tools. For example, e-mail has become one of the most common communication tools today, enabling users to discuss various project-related and non-project-related matters. Groupware enables the structuring of these activities by facilitating the means to aggregate e-mail related to one department into one repository and thus enable all authorized users within that department to view department-specific discussions. Groupware is thus a vital element of an organization's knowledge management

strategy, as it facilitates access to the intellectual capital and knowledge repositories of the enterprise.

Groupware in this phase enjoyed its own niche of business processes. Recent months, however, have witnessed the blurring of this niche as groupware applications merge with enterprise applications to provide an end-to-end automation of business processes. It is common to see groupware applications that tap into enterprise databases, just as the new enterprise e-commerce applications incorporate groupware functionality. The two camps seem to have joined hands to serve the customer collaboratively.

To illustrate the first case, consider a department within an organization that ventures to maintain its own budgetary process and implements a simple application based on groupware products from Novell or IBM/Lotus. The application incorporates basic rules for approvals, reviews, and so on, and serves the need of the department. The department can also implement hooks that enable the application to tap into the enterprise's ERP database to extract key budgetary values and numbers, and then uses those numbers in its own application. This is a typical example of a groupware application that relies on the ERP application database to obtain certain information.

On the other hand, consider a shopping Web site that offers customers the opportunity to collaborate and communicate with sales agents before buying a product. The Web site offers customers the opportunity to send e-mail to agents or collaborate through an electronic whiteboard to share information. In this case, the e-commerce application relies on groupware functions to offer a complete value-added solution to customers.

Groupware strengthens an enterprise's internal and external linkages by leveraging the Internet and other electronic channels. This facilitates ceaseless information flow through the enterprise and thus boosts the effectiveness and efficiency of business processes. To summarize, groupware products and services:

- Enable the setup of electronic communication channels between an organization's users and business partners.
- Establish a technology infrastructure that maximizes information sharing within the enterprise as well as across an enterprise's external linkages.
- Enable the streamlining of business process workflows not feasible or practical through other applications and systems (e.g., legacy and ERP systems).

- Provide an infrastructure for the management and sharing of unstructured information such as e-mail, discussions, and so on.
- Provide an infrastructure for the management and sharing of structured information such as documents.
- Provide the foundation for a knowledge management infrastructure, as covered in Chapter 4. Groupware solutions are easy to set up by departments whereas the knowledge management infrastructure discussed earlier requires extensive planning and involves global information sharing. These solutions provide enterprises with an entry into the knowledge management paradigm.

Groupware has the following three primary functions:

1. Group communications, enabling communication between teams and individuals;
2. Electronic information sharing, enabling groups to share information through electronic means;
3. Workflow, enabling the definition of work processes in an orderly manner.

The next section will elaborate upon each of those functions separately.

6.4.1 Group Communications

This section discusses the various communication methods enterprises can deploy to enable robust communication among parties internal and external to the organization. All communication methods discussed in this section are not necessarily part of a groupware product. The section discusses details regarding e-mail, voice communication, and new trends of unified messaging that enable a one-stop mechanism for storing and retrieving various types of messages.

6.4.1.1 Voice Communication

Voice communication continues to be the most popular and ubiquitous form of communication. Businesses rely heavily upon this form of communication. The PSTN infrastructure, value-added services offered by telecommunication organizations, and internal PBX-based infrastructure provides enterprises with enhanced communication services. These services include group conferencing, call waiting, and multiway calling.

With the popularity and increased proliferation of data networks, organizations are exploring opportunities to migrate PSTN-based voice communication services to data networks. IP telephony or Voice over IP (VoIP), as explained in earlier chapters, provides organizations with such opportunities. The primary applications of IP telephony, however, have involved the installation of telephony gateways that map between data and circuit-switched voice. True migration to IP telephony would rely solely upon data networks to fulfill an enterprise's voice communication needs. This would alleviate enterprises from the cost and administration headaches associated with maintaining two disparate networks. Although, as explained in Chapter 3, networking technologies have matured considerably to enable such migration, the industry is still wrestling with multiple issues to make IP-based voice communication mainstream. These issues include:

- Matching IP voice quality with circuit switched voice quality;
- Offering PSTN-like services (e.g., group conferencing, call waiting);
- Finalizing the cost models of telecommunication organizations for charging IP voice services;
- Infrastructure overhaul costs;
- Ubiquity of information appliances that would be as easy to use as the telephone.

The use of VoIP has generally been the greatest among freelance users on the Internet. Using tools such as Microsoft's NetMeeting, users communicate with friends, family members, and other users on the Internet. Recognizing this empowerment has triggered organizations to offer IP telephony-based services on their e-commerce sites, from customer service support to entertainment and other forms of content.

For the intraenterprise domain, VoIP implementations have started to surface in an effort to deliver a converged data and voice network to users in order to reduce an enterprise's costs associated with the maintenance of separate voice and data networks, but it is too soon to predict whether VoIP will replace the current methods of circuit-switched voice communication available through the telephone network. Thus far, few enterprises have ventured to use VoIP to enable their corporate users to communicate over the Internet or an intranet.

6.4.1.2 E-Mail

E-mail is the one of the most common and popular forms of computer communication, for business and nonbusiness users alike. The process of e-mail involves users formulating an e-mail message using a client e-mail application. The message then travels to a mail server that transmits the message through the Internet (or intranet) to the destined e-mail server. The recipient periodically signs-on to the mail server and retrieves the message sent by the sender. Various groupware products such as Novell's GroupWise, Microsoft Exchange, and Lotus's Domino deliver robust e-mail communication engines that support various e-mail formats. Some popular e-mail client applications are Netscape's Messenger, Microsoft Outlook, and Qualcomm's Eudora.

Figure 6.1 illustrates a simplified e-mail process flow. E-mail, much like other applications, has evolved through multiple protocols and standards. The following describes the more popular e-mail communication protocols and standards.

6.4.1.3 Simple Mail Transfer Protocol (SMTP)

SMTP is a TCP/IP-based mail transfer protocol over the Internet between one mail server and another. SMTP deals solely with the sending and receiving of e-mail messages. Once the mail reaches the mail server, POP3 or IMAP protocols (described later) deliver the message to the user's client application. The Internet standard RFC 821 specifies the SMTP protocol.

A UNIX application known as sendmail is the most popular SMTP implementation. However, recently many mail servers have surfaced that implement the SMTP protocol. Popular mail servers include Microsoft Exchange, Netscape Messaging Server, and HP's OpenMail.

6.4.1.4 Multipurpose Internet Mail Extensions (MIME)

The MIME protocol is an Internet standard that deals with the sending and receiving of mail in multiple data formats (besides plain ASCII text). The various formats include video, audio, graphics, and other formats. MIME works by

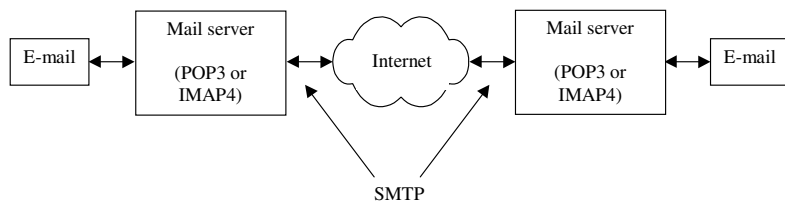


Figure 6.1 Internet e-mail architecture.

including special headers in the e-mail message. These headers indicate the type of message included in the data packet and enable client-side mail applications to load an appropriate application (player) to display or output the content of the e-mail message. Most client-side mail applications include support for popular data formats (e.g., GIF, JPEG, etc.) and automatically display the result of the message. For other data formats, the mail application loads another player application (similar to browser plug-ins) to display the contents of the message. The Internet standard RFCs 2045, 2046, 1047, and 2048 specify various details of the MIME protocol.

6.4.1.5 Post Office Protocol 3 (POP3)

POP3 is used by e-mail clients to retrieve e-mail messages from an e-mail server. POP3 is a client/server protocol that enables e-mail clients to retrieve and download e-mail messages from the server to the client. Therefore, POP3 defines the interface between the client application and the mail server and includes specifications for manipulating mailboxes on the user's client mail application. POP3 uses store-and-forward techniques that work by storing the mail on the server until the user signs on to the mail server, when the mail is forwarded to the client application. The Internet standards RFCs 1725, 1734, and 1082 specify various details of the POP3 protocol.

6.4.1.6 Interactive Mail Access Protocol (IMAP)

IMAP is an alternate mail protocol for accessing e-mail on the e-mail server. However, unlike POP3, IMAP provides the additional benefit of leaving e-mail messages on the e-mail server, rather than forcing the download to the client application. IMAP also extends features of the POP3 protocol to manipulate e-mail messages and mailboxes (folders, etc.) on the e-mail server. Using IMAP, users can view the sender's information and subject heading of the e-mail message and have the option to download the message. The Internet standard RFC 1730 specify various details of IMAP.

6.4.1.7 Unified Messaging

Enterprises usually implement multiple forms of message communication. These include e-mail, voice, paging, fax, and other types of messages. All forms of communication differ in the manner in which the messages are composed, delivered, stored, and retrieved. For example, users send e-mail messages using an e-mail client, whereas pager messages are composed either through the telephone interface or through a two-way pager device. This raises multiple issues. For example, to send messages, users need access to multiple information appliances to compose these messages (e.g., to send an e-mail message, users require

access to an e-mail client that connects to a mail server). Similarly, to send a voice message, the user calls a telephone number and leaves a message on the recipient's voice mailbox. Furthermore, to receive messages, users have to access those messages through various information appliances. Supporting these forms of messaging and communication protocols needs maintenance of different technology platforms and infrastructure. For the enterprise, this increases the total cost of ownership resulting from the cost of platforms as well as administration activities.

The unified messaging paradigm provides a universal platform for the storage of all types of messages in one user mailbox while providing a standard interface for the user to retrieve the messages in any desired format. For example, Bob can send an e-mail message to Susan that will be stored in the universal messaging repository. In the unified messaging model, Susan can retrieve the e-mail message through the e-mail application. Alternately, Susan can use her telephone and in conjunction with the unified messaging server's text-to-speech technology, hear the message through the telephone. For example, MailCall E-Mail Reader is an application by Phonesoft that allows the use of telephone to hear e-mail messages stored in a Lotus Notes database. Similarly, Unified MailCall is a complete voice mail system that allows the storage of voice mail messages in a Lotus Notes mailbox and allows user access through the telephone as well as the desktop. Lucent Technology's Octel Unified Messenger is another unified messaging solution that allows users to consolidate fax, e-mail, and voice messages in one server.

Various technology standards enable the building of unified messaging solutions. These technologies enable the integration of non-compatible technologies and data formats. For example, the Telephony Application Programming Interface (TAPI) enables the conversion between circuit-switched voice and IP data packets. Similarly, the Speech Application Programming Interface (SAPI) provides technologies for speech recognition and speech-to-text technologies.

6.4.2 Group Information Sharing and Collaboration

Internet collaboration refers to the live interaction between two or more users through their information appliances. Collaboration enables users to share information through chats (sending text messages), sharing documents, whiteboarding, and other means. Collaboration also enables users to work together using various scheduling and calendaring tools. The following delineates the popular collaboration and information sharing tools:

- *Group scheduling:* Group scheduling enables users to access a common repository to schedule meetings. Using group-scheduling tools, enterprise staff on the same enterprise groupware infrastructure can use their client applications to plan work activities, schedule meetings, and invite other users on the messaging network to those meetings. All invited users can check their names for the specific meetings through their groupware client applications (e.g., Microsoft Outlook). Various messaging servers (e.g., Microsoft's Exchange server or Netscape's Calendar server) provide the underlying infrastructure to connect all users and maintain their schedule information in one common repository. With the increase in mobility, users can use various information appliances to schedule such work activities. For example, an executive can sign on to the enterprise's central groupware server using his PDA (e.g., a Windows CE computer) and download schedules for meetings that his or her assistant has uploaded onto the server.
- *Information sharing:* Using a groupware client application, users can establish public folders and publish information to those folders, enabling users to share information. Users can also set up folders for which access is restricted to a certain group of people.
- *Discussion groups:* Groupware products also allow the setup of discussion groups that allow messages from various parties to appear on the discussion group window. Users can access these discussion groups using news-reader applications such as Netscape's Messenger or other groupware client applications such as Microsoft Outlook.
- *Maintaining a common contact database:* Groupware products enable users of a certain group to maintain a common contact list applicable and useful to users within that group. For example, the marketing department can maintain a list of customers. Setting up this list can enable all mobile sales staff to look up information through their mobile information appliances without sending e-mail messages to their department assistants or calling peers, thus tremendously saving time and increasing productivity.
- *Web collaboration:* Web collaboration refers to two parties sharing information over the Web. For example, Bob could send Web pages or Web links to Susan over the Web through a Web collaboration server that connects all collaborative parties over an intranet or the Internet. These solutions are very useful in business-to-consumer solutions in which call-center agents can up-sell and cross-sell solutions to their customers. Agents can also discuss the contents of various documents

with customers over the Web by looking at the same information (e.g., a utility bill) as the customer. Organizations can also use Web collaboration to display products to their consumers as consumers shop on their Web sites. For example, Lands End, Inc., a direct marketer for clothing and other products, enables consumers to interact with sales agents who can push specific products to consumers' Web browsers as consumers chat with the agents through the Lands End Web site. For intraenterprise solutions, businesses can also use these solutions for product demonstrations and training sessions. Weblines Communications, Inc. offers a suite of products that enable collaboration over the Internet and intranets.

- *Whiteboarding*: This feature enables networked users to share information on a common window that appears on all user's screens. All information posted to the screen either by typing on the window or cutting and pasting from other applications is visible to all users who are sharing the whiteboarding window. This feature enables the user to enter into discussions as if they were sitting in a common room. NetMeeting provides this feature and enables users to collaborate in this fashion over the Internet and intranets.
- *Application sharing*: This feature allows various users to share the execution of a program. Users can view the output of an application as it executes on another user's computer. In this scenario, the user who shares the application can control the application (e.g., terminate the application).

6.4.3 Workflow

Workflow provides intelligent sequencing and structure to activities that are not implemented through an enterprise's application systems. Workflow activities usually incorporate either nontransactional computer activities or functions that rely upon pure human intervention and cannot be computerized easily (e.g., reviewing the contents of a document, approving documents, verifying approver's signatures). In providing this structure, workflow tools leverage e-mail and other communication and information sharing tools. For example, by using a groupware product a department may define a specific workflow that would enable buyers (requisitioners) to enter an order through e-mail and route it to the supervisor. The supervisor, upon receiving the e-mail, would approve, disapprove, or request additional information from the requisitioner. The next workflow step (depending on the defined rules) may page the requisitioner if the supervisor requires additional information. A workflow product

therefore provides the structure to the ad hoc activities already being followed in the organization.

Workflow matches people with various activities and information. Workflow steps carry information and activities from one user to another and empower users to work effectively within each process step. As explained earlier, an enterprise's core business processes are usually implemented through its backend applications. For example, various business processes related to the supply chain are facilitated through business applications that reside on an organization's ERP or legacy systems. However, various work steps such as getting approvals and interacting with customers require employees to devise their own business processes and workflow. Groupware solutions enable users to structure those activities. A supply-chain system, for example, may generate a report through a batch system on orders that a department needs to process on a specific day. Instead of an employee signing on to the system to generate that report, an enterprise can devise workflow that would fax or e-mail the report to the employee. The workflow product would then track that activity and generate reminders for the employee if the orders were not processed in due time. After certain reminders, the groupware product would generate an alert for the supervisor on the status of those orders.

Workflow and business process design require graphical tools that can illustrate the entire process flow and facilitate additions, modifications, or deletions of various process steps. Development tools enable designers to design forms and define the various data elements on those forms. The development tools allow designers and administrators to define various users of the business process as well. And designers can define various rules for the workflow application (e.g., route the form to the supervisor for approval if the form data includes a purchase request of more than a specific dollar amount).

The new groupware products that provide these functions offer Web-enabled functionality, thus enabling enterprises to include external business partners and customers as part of the desired workflow. A central administrator can thus control the flow of the application across the Internet and include external parties' part of the work steps. Netscape Application Server: Process Automation Edition is an example of one such workflow engine. Lotus Domino Workflow is another example of a product that allows groups of users to define workflow and process for various activities.

The Workflow Management Coalition is a nonprofit organization of workflow vendors, users, analysts, and university/research groups whose mission is to formulate standards for various workflow products. The Workflow Management Coalition (WfMC) defines *workflow* as a series of steps during which documents, information, or tasks are passed from one participant to

another in a way that is governed by rules or procedures [12]. The organization cites improved efficiency, better process control, and business process improvement as being some of the primary benefits of workflow.

Enterprises can devise workflows of varying complexities. A workflow may incorporate simple steps that, for example, involve sending e-mail, and do not include any business rules. On the other hand, a workflow may interface with backend applications and require extensive application development. The following scenarios illustrate the complexity involved in devising the two types of workflow.

The following procurement process is quite simple and leverages the enterprise's e-mail infrastructure:

- A user sends e-mail to a supplier requesting a quote for a product.
- The supplier responds via e-mail (or telephone) with the quotation.
- The buyer sends e-mail to the supervisor requesting authorization.
- The supervisor sends e-mail indicating the approval.
- The buyer sends the request to the supplier, who then processes the order and e-mails the invoice to the department secretary.
- The secretary processes the invoice and routes it to the accounts department, which then issues a check to the supplier.

The technology infrastructure required for this scenario is minimal as the department can rely upon the traditional communication channels of e-mail and telephone to process transactions. The workflow also applies to situations that involve a limited number of transactions. However, as the number of transactions increase, the supplier will not be able to cope with the volume. Similarly, the accounts department will require a more streamlined system to issue and track payments.

To cope with increased transactions and transactions involving high dollar amounts, the enterprise can deploy a system that relies upon an e-commerce application, a set of groupware products, and a set of interfaces to backend legacy systems. For example, consider the following workflow:

- A user (requisitioner) signs-on to an intranet-based procurement application.
- The requisitioner browses the online catalog, selects the desired product, and enters the order on a form provided by the application.
- The system automatically validates the form by checking the data fields and routes the application to the requisitioner's supervisor.

- The supervisor receives the order through the e-mail system in his or her e-mail inbox.
- The supervisor approves or disapproves the order by typing the word *approve* or *disapprove* on the subject line.
- For approved orders, the e-mail system automatically routes the message to the e-commerce application's backend, which then generates an EDI order and routes it to the supplier's mailbox through the Internet.
- The supplier processes the order and routes a payment to the buyer's accounts systems, which, after proper authentication and authorization, issues the payment.
- The e-commerce application also generates an internal transaction and routes it to the enterprise's legacy system, which then updates the department's budgetary files.
- For disapproved orders, the e-mail system—through groupware functionality—triggers a pager message to the requisitioner indicating the reason for disapproval.
- At any time during this process, the requisitioner can sign on to the groupware database and check the progress of the workflow or status of approval. The requisitioner can sign on to the groupware database through various information appliances such as wireless telephones and PDAs (e.g., Lotus's Wireless Domino Access allows access to Notes databases from cell phones and PDAs).

Implementing the above process requires the following:

- Definition of the workflow through a groupware product.
- Definition of the roles of people (requisitioner, approvers, and security administrators who can define the various roles in the product).
- An e-mail system that interfaces with the groupware's workflow product and the e-commerce application.
- An e-mail system that incorporates functionality and rules to trigger pager messages to appropriate users.
- Data connectors that interface with the enterprise's legacy systems (e.g., Lotus provides Domino Enterprise Connection Services).
- Implementation of an e-commerce procurement application (e.g., Netscape's BuyerXpert).
- Interface of the e-commerce procurement application with the VAN that processes EDI transactions.

- Online validation of the purchase orders at the supplier's site.
- Issuance of the invoice to the buying organization (potentially through an EDI system or e-commerce application).
- Linking of the seller's application to the backend shipping application.

This list by no means provides all of the steps required to automate the procurement application. Rather, it is meant to illustrate the manner in which groupware applications can interplay with various enterprise applications in order to realize an end-to-end business process.

In conclusion, it is vital to understand that not all groupware products provide this degree of functionality. Some tools are better at providing communication and collaboration features, whereas others provide robust workflow functionality. Enterprises may have to implement various tools from various vendors to build a solution that fulfills the requirements of their business processes.

6.5 E-Commerce Application Development Standards

This section highlights the various industry frameworks and tools used for the development of backend e-commerce applications. For internal deployment of e-commerce applications, enterprises usually have three choices. First, an organization can use various off-the-shelf application suites to integrate and deploy e-commerce applications such as storefronts and simple information access services. Organizations can also acquire suitable vertical application suites related to banking, health care, sales force automation, and so on. And finally, organizations can use various development toolkits by industry players such as Microsoft, Netscape, and HP to develop specialized modules, interfaces, and services.

E-commerce application suites and backend ERP applications are sometimes not sufficient to deploy full-fledged e-commerce applications and systems. In such cases, organizations must develop the necessary business functionality and integrate with appropriate prepackaged e-commerce applications. Numerous technologies and programming frameworks have surfaced to enable the development of backend business logic and components. These technologies suit different business requirements. This section presents some of the technologies necessary to build backend business logic for e-commerce applications.

6.5.1 Java Servlets

Java servlets are programs developed in the Java programming language based on Sun's servlet API specification. Java servlets run on the server and extend the functionality of the Web server, just as applets extend the functionality of the browsers by providing an executable code for the client. Servlets usually host requests from clients and service those requests by either connecting to a database directly through database interfaces like JDBC or invoking other applications that service all or select portions of the user requests.

The industry considers the servlet model to be a prime candidate for replacing CGI-based scripts and programs. (CGI-based scripts are the primary means of invoking backend applications and access databases.) This is because servlets provide a more efficient paradigm for invoking backend applications and connecting to backend databases. For example, unlike CGI scripts, servlets do not require a separate process for each browser request and hence provide an efficient mechanism for invoking server-side programs, especially for handling higher-volume traffic. Java servlets also maintain consistent connections with backend databases and other programs. Servlets also have the ability to interface directly with enterprise databases, which enables enterprises to write simple and medium-sized interactive e-commerce applications. Due to their roots in the Java programming language, servlets are portable across various platforms and interoperate with a multitude of Web servers, as their APIs enjoy wide support.

Servlets run under a servlet engine that is usually included in an application server. Various vendors provide application servers that support execution of servlets. These application servers are either full implementation of Web servers that support servlet development and execution or provide plug-ins and add-ons to existing Web servers to support servlets. Full Web server implementations include Lotus Domino Go Webserver, Java Web server, and Konsoft Enterprise Server. The IBM WebSphere Application server provides an add-on for execution of servlets.

6.5.2 Enterprise Java Beans (EJB)

EJB is a Sun specification used for building portable application business logic. EJBs are similar to JavaBeans, which are reusable client-side components used to develop information-appliance GUIs and other client-side application logic. EJB's extend the same model to the server side and provide a framework for the development of reusable server-side application components.

While servlets are ideal for developing simple e-commerce applications, the development of complex business logic requires a computing paradigm such as the one offered by EJBs. The EJB specification enables the development of object-oriented application modules and masks numerous development complexities from the developers, including transactional complexities. For example, EJB includes the specification of “entity beans” that enable developers to focus on developing business components and logic without regard to the underlying logic of accessing databases and transaction services. By coupling entity beans with databases, developers can write applications against entity beans. This makes EJBs portable across platforms and developers do not have to write SQL to access databases directly.

EJBs require “containers” that provide services for their execution and management in a runtime environment. Application servers usually provide these containers, which are similar to servlet engines, to support the execution of EJBs. Any application server that implements the EJB specification can support EJBs. This facilitates the porting of EJBs from one application server to another without any modifications. This is unlike the current implementation of enterprise transaction-based applications, which are locked into one vendor’s implementation so that porting application logic from one server to another requires extensive modifications. For example, an application component that is developed to run in a CICS environment requires extensive modification before it can run in BEA Tuxedo’s application environment. Legacy application environments such as CICS and Tuxedo do not interoperate with EJBs since they do not implement the EJB specification. However, their vendors have recently released new versions of their products that support such interoperability.

Three versions of the EJB specification have surfaced as of this writing: 1.0, 1.1, and 2.0. Application servers have implemented specifications through version 1.1 with plans to implement version 2.0. The 2.0 specification packs more functions for robustness, performance, and scalability.

Various development tools from vendors enable the development of EJBs. These tools are either standalone tools or are IDE as part of application servers. Examples of application development tools are IBM’s Visual Age, Symantec’s Visual Café, Inprise’s JBuilder, and BEA’s WebLogic.

6.5.3 Active Server Pages (ASP)

ASPs are HTML files that contain references to various scripts and programs that execute on the server when the user requests the ASP page. Upon executing an ASP page, the Web server produces an HTML file and sends it to the

Web browser. The server-side programs are usually VBScript and JScript programs that handle user input from a browser and can perform various operations such as accessing backend databases, retrieving results, and presenting the output to the user's browser as an HTML page on the appropriate information appliance. ASP files have the .asp file extension.

6.5.4 Java Server Pages (JSP)

JSPs are similar to ASPs. JSPs control the appearance of HTML pages on the user browser. JSPs have embedded Java code (Java servlets) that run on the server when accessed. Within the HTML file, appropriate HTML tags contain bracketed Java code. This approach differs from the one where backend Java programs runs to generate HTML code and sends it to browsers on information appliances.

Notes and Web Sites

- [1] www.omg.org/cgi-bin/doc?formal/98-12-09
- [2] www.iona.com
- [3] www.javasoft.com
- [4] www.novell.com/corp/strategy/fsd/whatis.html
- [5] Microsoft Exchange Server, Microsoft's vision of unified messaging white paper.
- [6] Management of Internet Names and Addresses, United States Department of Commerce, Docket number 980212036-8146-02.
- [7] Status Report to the Department of Commerce, June 15, 1999.
- [8] www.icann.org
- [9] www.icann.org/registrars/icann-pr21apr99.htm
- [10] www.corenic.org
- [11] Internet Governance, ICANN and ASI information (www.arin.net).
- [12] www.wfmc.org