

shopnovell

When the Going Gets Tough,
the Tough Go shopnovell

Cheryl Walton

Although there were no search lights to draw people in, shopnovell's grand opening was well attended. Novell opened shopnovell—its online store—at BrainShare '99 in Salt Lake City on March 21–26. If you are among the 6,002 networking professionals who attended BrainShare '99, you already know about shopnovell. If you did not attend BrainShare '99, you may know about shopnovell if you have recently visited Novell's web site (<http://www.novell.com>) and clicked the How To Buy button on Novell's service bar. After you click this button, you are presented with several links—including a shopnovell link—through which you can purchase Novell products. (You can also access shopnovell directly at <http://shop.novell.com/shopnovell>.)

shopnovell is more than a convenient way to purchase the latest Novell products, however. shopnovell is also a working example of how NetWare 5 serves as a solid foundation on which you can build your company's e-commerce system. That is, if your company is planning to join the growing number of companies doing business over the Internet, shopnovell is a case study demonstrating the advantages of using NetWare 5 to implement an e-commerce system.

SPEED SHOPPING

Over the years, Novell has refined its renowned NetWare file and print services. These refinements enable NetWare 5 to deliver file and print services at the unprecedented rate of up to 253.3 Mbit/s—68.8 Mbit/s faster than Windows NT 4.0 can deliver these services. (For more information about the tests upon which these rates are based, see the KeyLabs Inc. report entitled "NetWare 5 vs. Windows NT Server 4.0" at <http://www.keylabs.com/results/novell/nw5.htm>.) As Kevin Millecam, senior manager for Novell Electronic Marketing, points out, these refinements make NetWare 5 "an extremely efficient engine for serving up any kind of pages."

Taking advantage of NetWare 5 file services, Novell uses NetWare 5 as the platform for a bank of HTTP accelerator servers. (See Figure 1 on p. 25.) Novell then runs BorderManager FastCache Services 3 on these NetWare 5 servers and configures BorderManager as a reverse HTTP proxy server. This configuration enables BorderManager FastCache Services 3 to retrieve and return web pages and image files that are stored on shopnovell HTTP and HTTP Secure (HTTPS) servers.



The HTTP and HTTPS servers are responsible for creating and maintaining shopnovell web pages and are NetWare 5 servers running Netscape Enterprise Server for NetWare. (See Figure 1 on p. 25.)

Although BorderManager FastCache Services 3 must retrieve dynamic web pages and images each time a browser requests such pages, BorderManager FastCache Services 3 stores shopnovell's static web pages and image files in its own cache. Running on NetWare 5, BorderManager FastCache Services 3 can deliver static (and therefore cached) web pages at up to 500 Mbit/s. BorderManager FastCache Services 3 also allows customers to access shopnovell at rates of up to 10,000 hits per second.

In addition, BorderManager FastCache Services 3 saves Novell the time and expense of implementing a separate security system for shopnovell. The BorderManager servers provide the only point of access to shopnovell. Behind this point of access, shopnovell web, application, and database servers operate unseen by—and therefore beyond the reach of—customers' browsers. Millecam comments on BorderManager's role in making shopnovell secure, "BorderManager takes care of all the security. We didn't have to worry about writing some kind of ad hoc security scheme in addition to everything else."

A LITTLE SHOPPING, A LITTLE JAVA

Many of the services that shopnovell HTTP and HTTPS servers provide run on the Novell Java Virtual Machine (JVM) for NetWare. (The Novell JVM enables you to run Java programs. In other words, the JVM is like an operating system running on top of the NetWare 5 operating system.) The most obvious way shopnovell exploits the Novell JVM is through Evergreen Internet Inc.'s ECential. ECential is a Java software package that runs on the Novell JVM and provides e-commerce

capabilities for shopnovell. (For more information about Evergreen and ECential software, visit <http://www.evergreen.com>.)

In addition, shopnovell takes advantage of the following features provided by the Novell JVM:

- The Java servlet Application Program Interface (API) enables servlets to access information from both Java and non-Java programs running on the NetWare 5 server.
- The Java Database Connectivity (JDBC) API allows Java programs, such as servlets, to access information stored in database applications, such as Oracle.

For example, a servlet might use the Java servlet API to obtain database information through the JDBC API. (For an explanation of a servlet, see the glossary on the *NetWare Connection* web site at <http://www.nwconnection.com>. For more information about the Novell JVM, see "Wake Up to Server-Side Java," *NetWare Connection*, June 1999, pp. 6–18. You can download this article from <http://www.nwconnection.com/past>.)

Java Servlet API

shopnovell takes advantage of the Java servlet API through several ECential servlets that run on the HTTP and HTTPS servers. These servlets enable shopnovell customers to use their web browsers to communicate with and to transact business with shopnovell. For example, two ECential servlets create and maintain the shopping cart pages and shopnovell product pages through which customers can purchase products on shopnovell.

When a shopnovell customer requests a dynamic web page, such as a shopping cart page, BorderManager FastCache Services 3 passes the request to the ECential servlet responsible for creating that page. Servlets that create dynamic web pages for shopnovell act indirectly as liaisons between a customer's browser, the NetWare application servers, and the servers upon which shopnovell databases reside.

For example, when a customer clicks the View My Shopping Cart bar, which is located at the bottom of the shopnovell catalog pages, the following events occur:

- BorderManager FastCache Services 3 sends the request to the shopping cart servlet.
- The shopping cart servlet accesses the shopping cart service, an ECential component that runs on the NetWare 5 Java/Common Object Request Broker Architecture (CORBA) application server. (See Figure 1 on p. 25.)
- The shopping cart service accesses information from the Oracle database.
- The shopping cart servlet uses Java Server Pages (JSP) to create a web

DID YOU KNOW . . .

Online shopping means big bucks for information technology (IT) companies. In fact, four IT companies—General Electric Co., Cisco Systems Inc., Intel Corp., and Dell Computer Corp.—plan to collectively generate more than U.S. \$35 billion in online sales by the end of this year. (See the U.S. Government Working Group on Electronic Commerce First Annual Report, Nov. 1998. You can download this report from <http://www.doc.gov/ecommerce/review.htm>.)

page that contains the information about the customer's shopping session. (For an explanation of JSP, see "Java Server Pages" in the glossary on the *NetWare Connection* web site at <http://www.nwconnection.com>.)

- The shopping cart servlet sends the web page back to one of the BorderManager FastCache Services 3 servers.
- The BorderManager FastCache Services 3 server sends the web page to the customer's browser.

CORBA and the JDBC API

Before requesting information about the contents of a shopping cart, a cus-

tommer usually clicks through the shopnovell catalog and adds products to the shopping cart. The product page servlet responds to a customer's request for information about a particular product and delivers that information in the form of a dynamically created web page. Like the shopping cart servlet, the product page servlet must access another application to obtain the necessary information.

How do these two servlets communicate with applications running on other servers? Both servlets communicate with other applications via CORBA and the JDBC API. The shopping cart servlet and the product page servlet use CORBA to access other components of the ECential e-commerce software. (CORBA allows an application running on one server to communicate with an application running on another server. For more information about CORBA, see "The Ins and Outs of CORBA" on p. 28. For more information about how shopnovell uses CORBA to enable communication between applications, see the "shopnovell—Open for Business" section on p. 25.) Both servlets also use CORBA in conjunction with the JDBC API to access the Oracle database in which shopnovell stores product information.

ONE, TWO, THREE, FOUR, CAN I HAVE A LITTLE MORE?

The JDBC API allows Java programs—such as the product page servlet—to access and manipulate data through Structured Query Language (SQL) statements and commands. Most relational database management systems, including Oracle, support SQL, and by extension these databases also support the JDBC API. (For more information about SQL, visit <http://www.geocities.com/ResearchTriangle/Node/9672/sqltut.html>. For a concise explanation of SQL and relational databases, see the glossary on the *NetWare Connection* web site at <http://www.nwconnection.com>.)

The JDBC API also supports distributed databases, such as the Novell Directory Services (NDS) database, which contain files located on more than one computer. Distributing any kind of database, including a relational database, enables more than one user or program to access that database at any given time and also decreases the workload that any one computer must support.

In the future, Novell plans on leveraging this support for distributed databases by implementing Novell Cluster Services for NetWare 5 to protect the database that stores shopnovell product information. (See Figure 1.) Currently in open beta, Novell Cluster Services for NetWare 5 is software that guards against hardware and software failure. With Novell Cluster Services for NetWare 5, you will be able to provide fault tolerance for applications that are essential to your company's operations.

Novell Cluster Services for NetWare 5 will allow the shopnovell Oracle database servers to share information that is stored on a shared physical disk drive using SCSI. Through Novell Cluster Services for NetWare 5, the shopnovell Oracle database servers will be able to monitor each other's ability to manage shopnovell data. Should one of these database servers fail, explains Thomas Lonni, shopnovell project manager, "clustering technology will allow the other [server] to immediately pick up the slack so the site doesn't go down." (You can download the open beta version of Novell Cluster Services for NetWare 5 at <http://support.novell.com/beta/public>.)

Novell Cluster Services for NetWare 5 will also enable shopnovell administrators to balance the workload among the shopnovell Oracle database servers. In addition, shopnovell administrators will be able to add database servers to the cluster as more database servers are required to accommodate shopnovell's growth.

SHOPNOVELL—OPEN FOR BUSINESS

The Java servlet API and the JDBC API, which are included with the Novell JVM, enable communications between Java programs and relational databases. However, shopnovell, like many e-commerce systems, includes several types of applications that must communicate with one another. In addition, shopnovell, like most e-commerce systems, must be able to grow as demands on the system grow. VisiBroker for NetWare enables shopnovell to meet both of these requirements.

VisiBroker for NetWare is a CORBA-compliant Object Request Broker (ORB) that is based on Inprise Corp.'s VisiBroker for Java 3.3. (For more information about VisiBroker for NetWare, visit <http://developer.novell.com/ndk/doc/orb>.)

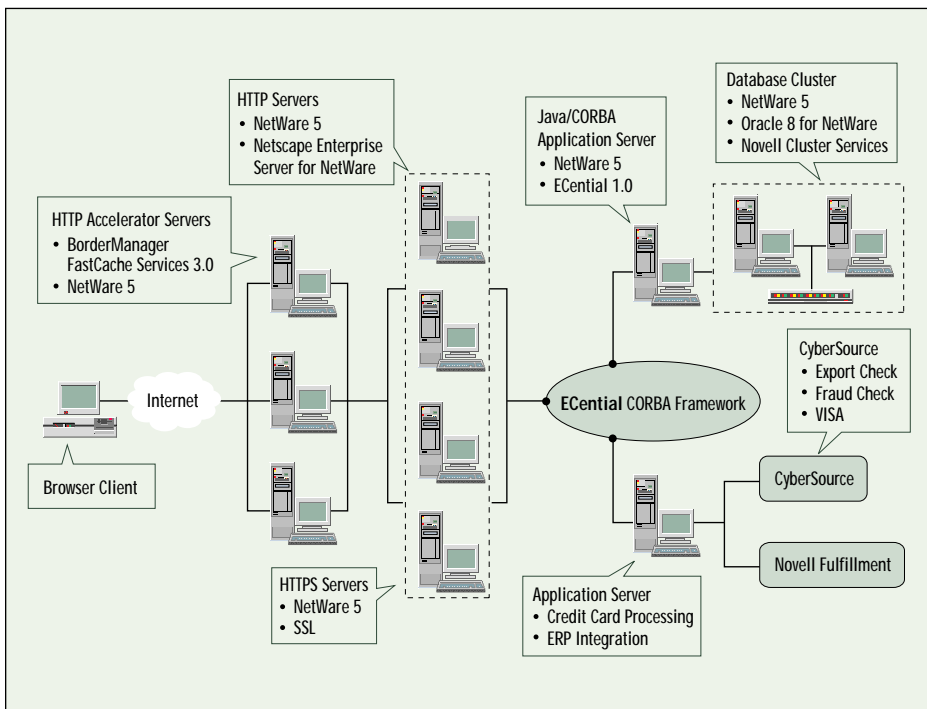


Figure 1. The shopnovell architecture. BorderManager FastCache Services 3 servers act as a liaison between the customer's browser and shopnovell.

Prosoft Engineering 1/3 Page AD

4 7/8" x 4 7/8"
(4.875" x 4.875")

The Ins and Outs of CORBA

Imagine a middleman who brokers objects from all over the world. If a retailer sends this middleman a request for a particular object, the middleman can quickly locate the manufacturer that can fulfill this request. No matter what language the retailer uses or the manufacturer uses, this middleman can communicate to the manufacturer the exact specifications of the object the retailer is requesting. In addition, the middleman can advertise objects on behalf of the manufacturer so that retailers know what objects are available. The middleman can also quickly and effectively transport requested objects from the manufacturer to the retailer.

Just as this hypothetical middleman helps businesses that may not otherwise be able to communicate with one another, Common Object Request Broker Architecture (CORBA) helps applications that might not otherwise be able to communicate with one another. CORBA is middleware—the software equivalent of a middleman—through which an application can request pieces of information (or objects) from another application. Like the hypothetical middleman, CORBA enables one application, called the client, to request pieces of information (or objects) from another application, called the object implementation. In short, CORBA takes a client's request and finds an object implementation that can fulfill that request. For example, CORBA can enable a client to request pricing information from an object implementation that is able to supply that information.

THE BARE BONES

How does CORBA intermediate between two applications, regardless of the applications' language differences? Programmers can use a variety of CORBA interfaces to enable clients to access information from object implementations. For example, a programmer can use CORBA Interface Definition Language (IDL) stubs and IDL skeletons to enable interapplication communication. (See Figure 4 on p. 31.)

IDL is a language that defines the types of objects that applications can request from CORBA. IDL also defines the specific operations that can be performed on these objects. IDL stubs are interfaces through which a client can request CORBA services. IDL stubs map client requests from the client's language to IDL, the language CORBA understands.

IDL skeletons are interfaces through which CORBA communicates with object implementations. IDL skeletons map IDL to the object implementation's language.

Programmers create IDL stubs and skeletons by writing an IDL specification for each type of object a client can request from an object implementation. For example, if a programmer wants to enable a Java application (the client) to access information from a C++ application (the object implementation), the programmer writes an IDL specification for each type of C++ object the Java application must access. These C++ objects could include information such as product pricing information.

These IDL specifications will also include the types of operations that can be performed on the objects. After writing the necessary IDL specifications, the programmer runs these specifications through a CORBA IDL compiler. The compiler then produces IDL stubs written in the Java language for the client and IDL skeletons written in C++ for the object implementation. Finally, the pro-

grammer adds the IDL stubs to the client and the IDL skeletons to the object implementation.

THE DYNAMIC OPTION

Another way programmers can enable CORBA to intermediate between applications is through a Dynamic Invocation Interface (DII) and Dynamic Skeleton Interface (DSI). A DII enables CORBA to accept client requests for services, and a DSI enables CORBA to access object implementations. (See Figure 4 on p. 31.)

Through a DII, a programmer can enable a client to specify the following:

- The type of object the client needs
- The operation the client wants CORBA to perform on that object
- Any special parameters that must be applied to the object

The code that allows a client to request CORBA services through a DII resides in CORBA's Interface Repository. The Interface Repository is a database in which language-specific mappings to CORBA objects and operations reside.

Through a DSI, a programmer can enable an object implementation to respond to client requests for objects. The code that allows an object implementation to supply CORBA services to a requesting client also resides in CORBA's Interface Repository.

THE SPHERE

CORBA also provides services through Object Adapters and the Object Request Broker (ORB) Interface. Object Adapters are software components that interface directly with the ORB core. (See Figure 4 on p. 31.) These components allow object implementations to access the core services CORBA provides, such as mapping object references from the object implementation's language to IDL.

The ORB core comprises the infrastructure upon which all other CORBA components are built. In addition, the ORB core provides CORBA with the basic communication services necessary to transfer requests from clients to object implementations. The ORB Interface provides both clients and object implementations with direct access to these basic services.

In addition, CORBA can advertise available objects and services on behalf of object implementations. CORBA advertises these objects and services via a Common Object Services Specification (COSS) service called the trader service. (For more information about COSS, download "A Discussion of the Object Management Architecture" from <http://www.omg.org/library/oma/oma-all.pdf>.)

The hypothetical middleman uses advertising to let retailers know about new objects that are available for the middleman to broker. The more objects the middleman brokers, the larger his or her business can become. Similarly, by taking advantage of the CORBA trader service's ability to advertise, programmers can enable their systems to grow. The CORBA trader service allows programmers to add new object implementation application servers to a system whenever and wherever those servers are needed.

The CORBA trader service also enables clients to access the services offered through these new object implementation servers without further ado. In fact, CORBA is to the computing world of clients and object implementations what the hypothetical middleman is to his or her world of retailers and manufacturers, with one notable exception: CORBA isn't hypothetical. ●

Before you can understand how Visi-Broker for NetWare meets shopnovell's needs, you must understand the following terms:

- ORB. An ORB allows one application to access information from another application.
- CORBA. CORBA is an Object Management Group (OMG) standard that defines programming interfaces to the OMG implementation of an ORB. (For more information about CORBA, see "The Ins and Outs of CORBA" on p. 28. For more information about the OMG, visit <http://www.omg.org>.)
- CORBA/Internet Inter-ORB Protocol (IIOP) specifications. The OMG, a consortium composed of more than 800 IT companies (including Novell), is responsible for developing the CORBA/Internet Inter-ORB Protocol (IIOP) specifications. (IIOP is a protocol that transports CORBA over an intranet or the Internet.)

CORBA operates as the lingua franca, or common language, by which a CORBA-enabled application can speak to, and transact business with, other CORBA-enabled applications. As Chris Staszak, former chief technology officer for Evergreen, explains, "CORBA doesn't care if the application sitting on the server is written in COBOL, C++, or Java." Whether a Java-based e-commerce system includes a legacy enterprise resource planning (ERP) application written in COBOL or a credit-card authorization application written in C++, CORBA allows the various components of the e-commerce system to communicate with one another. "With so many technologies, there's a lot of hype, [but] with CORBA all the good things about it are real," adds Staszak.

CORBA enables the servlets running on the web servers and other ECential components running on the NetWare 5 Java/CORBA application server to interact with one another. (See Figure 1 on p. 25.) In fact, CORBA is the language that allows all of the applications running on each of the servers in the shopnovell network, including the database servers, to communicate with one another.

Providing a common language through which different applications can communicate with one another is certainly

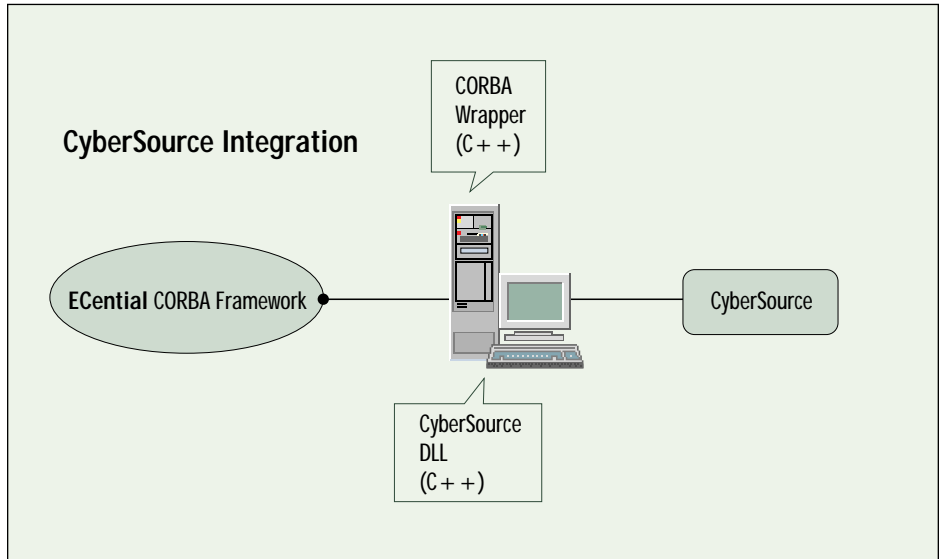


Figure 2. shopnovell exchanges credit card information and electronic delivery information directly with CyberSource in Santa Clara, California.

one of the more important services CORBA provides for shopnovell, but it isn't the most important service. The most important service CORBA offers is the ability for object implementations to advertise their services. Object implementations are the applications that respond to CORBA's request for information. Because CORBA enables object implementations to advertise their services, Novell can add servers to shopnovell wherever these servers are needed.

For example, when the product page servlet broadcasts its request for information over the shopnovell network, the product service running on the NetWare 5 Java/CORBA application server broadcasts back its ability to provide the requested services. If an additional application server were added to the shopnovell network, the product service running on this additional server would also advertise its ability to respond to the product page servlet's request. In other words, says Lonni, as shopnovell grows, all Novell needs to do "is take a new server and add it in the system, and because of CORBA, shopnovell will automatically start using that new server."

CREDIT CARDS ACCEPTED

In addition to enabling applications within shopnovell to communicate with one another, CORBA enables shopnovell applications to communicate with CyberSource applications running outside of shopnovell. (See Figure 1 on p. 25.) CyberSource is a company that provides

shopnovell with credit card services—such as credit card preauthorization and billing—as well as with electronic delivery of Novell products.

NetWarefiles.com
1/6 Page
AD

2 3/8" x 4 7/8"
(2.375" x 4.875")

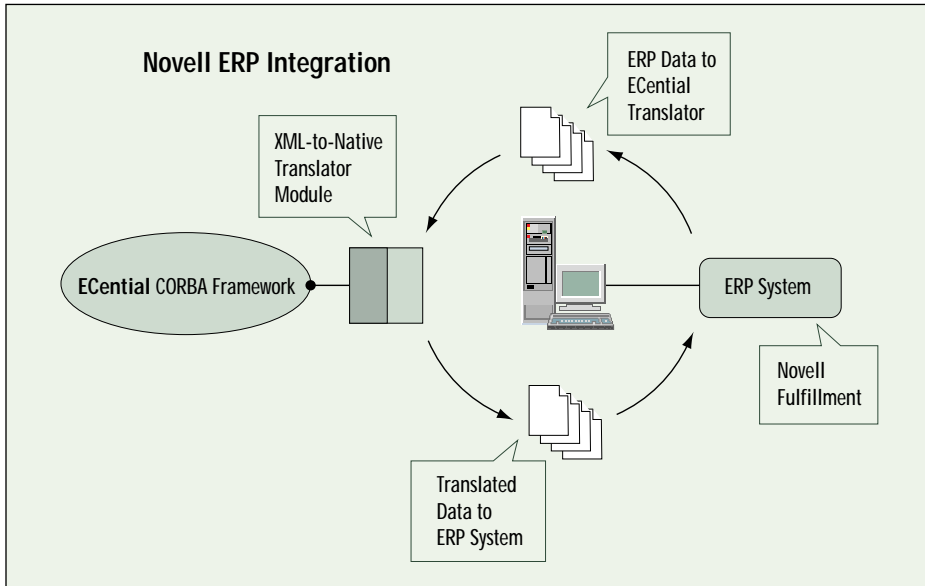


Figure 3. Communications between shopnovell and the fulfillment house's Enterprise Resource Planning (ERP) system are made possible through an XML-to-native translator module.

shopnovell orders are delivered to customers in one of two ways, depending on the particular product. Products can be delivered either as an electronic download via CyberSource Sm@rtCert digital delivery or as a physical product via United Parcel Service (UPS) ground delivery. (For more information about shopnovell delivery options, visit http://shop.novell.com/shopnovell/help.html#delivery_options.)

When a customer enters his or her credit card information and clicks to submit an order, a servlet running on one of the HTTPS servers uses the Java servlet API to invoke Secure Sockets Layer (SSL). (For an explanation of SSL, see the glossary on the *NetWare Connection* web site at <http://www.nwconnection.com>.) The credit-card servlet then relays the customer's information via CORBA/IIOP to the shopnovell application server running a CyberSource application. (See Figure 1 on p. 25.) This application server uses the customer's credit card information to formulate a specialized request for credit card services, which the server then sends to CyberSource in Santa Clara, California via a direct line. (See Figure 2 on p. 29.)

After receiving the request for services from shopnovell, applications running on the CyberSource network use the customer's credit card information to contact the appropriate credit card company for preauthorization. CyberSource

then returns the status of the customer's purchase to the application server running on the shopnovell network.

The credit-card servlet retrieves the customer's purchasing status from the application server and, via JSP, creates a web page on the fly that either confirms the customer's purchase or tells the customer that the purchase has been denied. Finally, the credit-card servlet passes this newly created web page to a

If the customer's credit card purchase is authorized, a shopnovell component sends an e-mail message to that customer confirming his or her purchase.

server running BorderManager FastCache Services 3, which uses port 443 to send the web page across the Internet to the customer's browser. (Port 443 is the HTTP accelerator server port responsible for transporting information via SSL.)

In addition, if the customer's credit card purchase is authorized, a shopnovell component running on an application server creates and sends an e-mail message to that customer confirming his or her purchase from shopnovell.

A FULFILLING EXPERIENCE

The application that formulates credit-card processing requests also receives information from the shopping cart service. When a customer purchases one or more products that can be delivered electronically, this application uses information from the customer's shopping cart to formulate another specialized request. (See Figure 2 on p. 29.) Like the request for credit-card processing, this request is sent to CyberSource. However, this request goes to Sm@rtCert, the CyberSource electronic warehouse/delivery application that is responsible for processing customer orders.

The Sm@rtCert application uses information contained in the request, such as product Stock Keeping Unit (SKU) numbers, to create an electronic certificate. (Sm@rtCert can deliver up to 15 separate SKU numbers.) This certificate gives the shopnovell customer the right to download the product(s) he or she has purchased. After creating this certificate, the Sm@rtCert application returns the certificate number to the shopnovell application server that originally sent the request.

After receiving the certificate number, the requesting application forwards that number along with the Sm@rtCert warehouse URL to the ECential servlet that creates order acceptance pages for shopnovell customers. This servlet includes the certificate number and the URL in the web page it creates and passes this page to one of the BorderManager FastCache Services 3 servers for delivery to the customer.

The customer who receives this information can then enter the Sm@rtCert URL in his or her browser and follow the directions for downloading Sm@rtCert client software. This client software, called the *Download Manager*, is a small program that downloads quickly.

The Download Manager enables a customer to locate his or her particular transaction and then to safely download the product(s) that he or she has purchased. If the customer's connection fails during the process of downloading his or

her newly purchased software, the Download Manager automatically re-establishes that connection. (For more information about Sm@rtCert electronic delivery, visit http://shop.novell.com/shopnovell/smartcert_instructions.html.)

After the customer has successfully downloaded his or her purchase, a shopnovell component running on an application server notifies CyberSource. CyberSource then charges the customer's credit card account for the amount of purchase.

Of course, not all shopnovell products are delivered electronically. How does shopnovell fulfill requests for non-electronically delivered products? To fulfill these requests, shopnovell communicates with another fulfillment house via its Enterprise Resource Planning (ERP) system. (For an explanation of the ERP system, see the glossary on the *NetWare Connection* web site at <http://www.nwconnection.com>.)

To communicate with this ERP system, one of the shopnovell ECential components, the order processor, translates a customer's order information into the particular language this ERP system requires. The order processor then formats this translated information and delivers it to the ERP system. (See Figure 3 on p. 30.)

After receiving this information, the fulfillment house processes the purchase order, ships the ordered product(s), and then returns to shopnovell the status of the customer's order (shipped) along with a shipping number. shopnovell customer service representatives can then access this information for customers who have inquiries regarding their orders. Beginning July 15, 1999, customers can also access information regarding their orders through the shopnovell site.

When shopnovell receives confirmation that a customer's order has been shipped, shopnovell automatically notifies CyberSource, which contacts the customer's credit card company for payment.

CONCLUSION

If you haven't been considering the best way to put your company's business online, you might soon be making this decision. According to the U.S. Government Working Group's First Annual Report, the number of people connected to the Internet—and therefore the po-

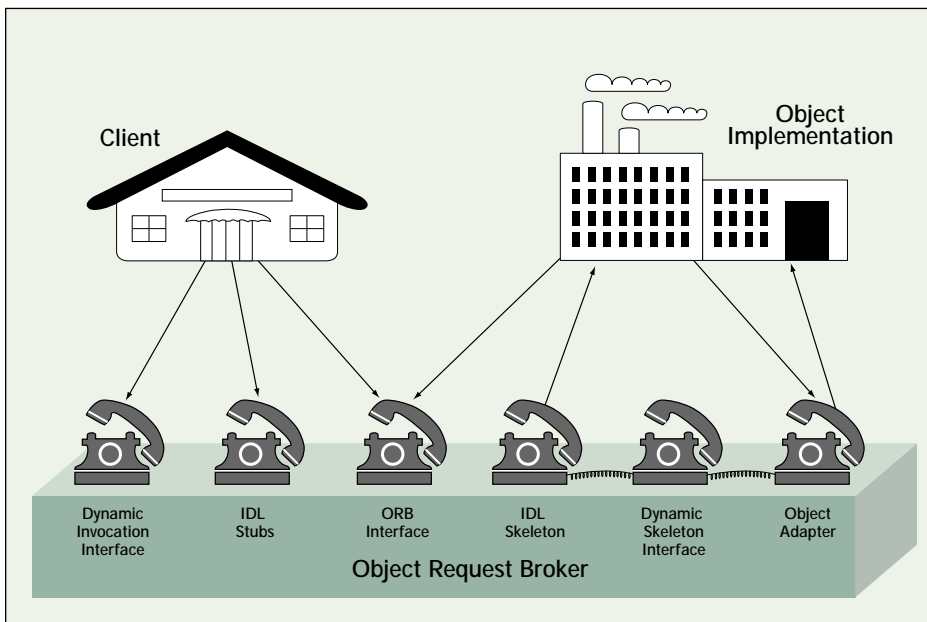


Figure 4. Applications running on shopnovell communicate via Common Object Request Broker Architecture (CORBA).

tential number of customers for online businesses—is increasing dramatically. When the Working Group released its report in November 1998, more than 140 million people were connected to the Internet—up from 50 million people in July 1997.

In addition, conducting business online may help increase your company's productivity. According to the Working Group, companies of all sizes are "realizing substantial productivity improvements" as a result of implementing e-commerce systems. (You can download this report from <http://www.doc.gov/ecommerce/review.htm>.) Needless to say, improvements in a company's productivity generally translate into improvements in that company's bottom line.

Also needless to say, an e-commerce system that builds on the system a company already has saves that company the cost of retooling its existing system. As Staszak puts it, replacing your company's IP infrastructure can be "cost prohibitive" at best. By building shopnovell on NetWare features such as the Novell JVM for NetWare and VisiBroker for NetWare, Novell has demonstrated how you can avoid the costs associated with retooling your company's infrastructure. In other words, if sometime in the future, you find yourself considering the best way to put your company's business online, let shopnovell be a lesson to you.

Cheryl Walton works for Niche Associates, an agency that specializes in technical writing and editing. ●

Atlantis
1/6 Page
AD

2 3/8" x 4 7/8"
(2.375" x 4.875")

For more information, visit
<http://advertise.nwconnection.com>.