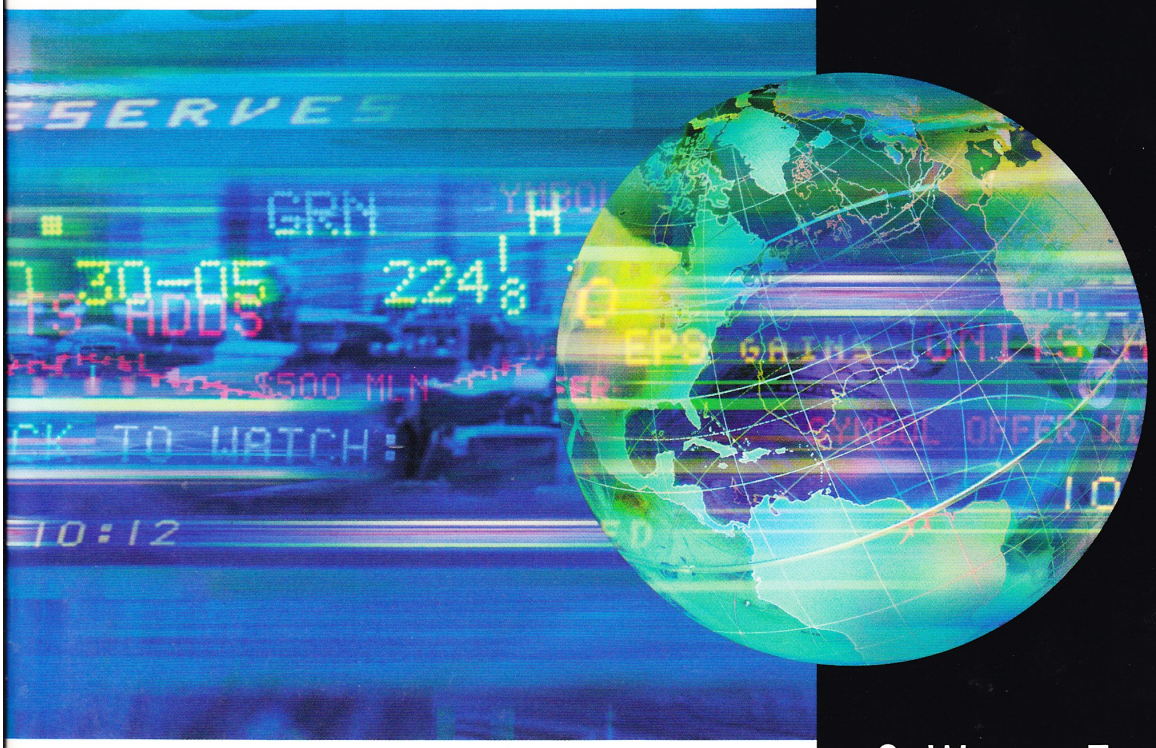
Designing Systems for Internet Commerce

Second Edition



G. Winfield Treese
Lawrence C. Stewart

APPL-1008
APPLE INC. / Page 1 of 20

Designing Systems for Internet Commerce

Second Edition

G. Winfield Treese
Lawrence C. Stewart

Addison-Wesley

Boston • San Francisco • New York • Toronto • Montreal
London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers discounts on this book when ordered in quantity for bulk purchases and special sales. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact:

International Sales
(317) 581-3793
international@pearsontechgroup.com

Visit Addison-Wesley on the Web: www.awprofessional.com

Library of Congress Cataloging-in-Publication Data

Treese, G. Winfield.
Designing systems for Internet commerce / G. Winfield Treese, Lawrence C. Stewart.—2nd ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-201-76035-5 (pbk. : alk. paper)
1. System design. 2. Electronic commerce. I. Stewart, Lawrence C. II. Title.

QA76.9.S88 T74 2003
658.8'00285'4678—dc21

2002074788

Copyright 2003 by Pearson Education, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher. Printed in the United States of America. Published simultaneously in Canada.

For information on obtaining permission for use of material from this work, please submit a written request to the following address:

Pearson Education, Inc.
Rights and Contracts Department
75 Arlington Street, Suite 300
Boston, MA 02116
Fax: (617) 848-7047

ISBN: 0-201-76035-5
Text printed on recycled paper
1 2 3 4 5 6 7 8 9 10—CRS—0605040302
First printing, September 2002

CHAPTER 21

Putting It All Together

> Three things are to be looked to in a building: that it stand on the right spot; that it be securely founded; that it be successfully executed.
>
> —Johann Wolfgang von Goethe[1]

Building Complete Systems

In Part One of this book, we discussed issues of business, architecture, and implementation. In Part Two, we discussed implementation technologies for various aspects of Internet commerce. In this chapter, we try to assemble the pieces into complete systems. There are many workable ways to build Internet commerce systems; the approaches we describe here are but a few examples.

We will first present an architecture for a federated commerce system. Next, we describe a core system architecture and its requirements, followed by a discussion of how this architecture may be applied to the business models discussed in Chapter 4.

Federated Commerce System

We envision many individual Internet commerce systems as being members of a global federated commerce system, as shown in Figure 21-1. In the federated commerce system, customers may join one or more buyer communities and conduct commerce

1. Johann Wolfgang von Goethe, *Elective Affinities* (1808).

APPL-1008
APPLE INC. / Page 4 of 20

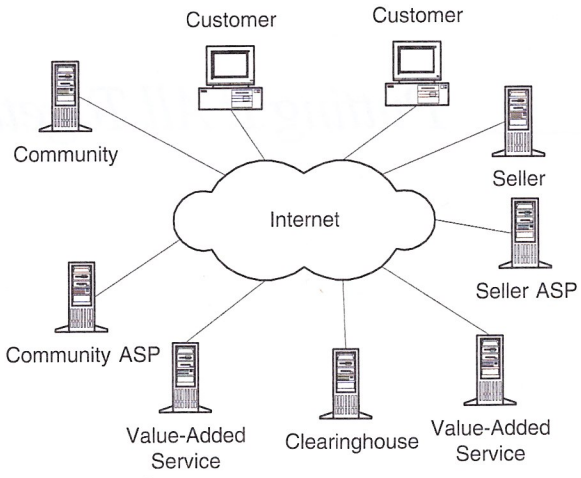


FIGURE 21-1. Federated Commerce System

with any number of sellers. Buyers get the advantages of a single account to remember and unified customer service. Sellers gain the advantages of standardization, ease of use, access to a large aggregation of customers, and easy access to network-based value-added services. In this example, we will abbreviate the generic *federated commerce system* as *FCS*.

The authors first proposed an FCS in a series of memoranda at Open Market in March 1995. In December 1998, Open Market launched an effort called *CommerceTone* to bring a federated commerce environment to the market. The project wasn't finished, but a separate company called CommerceTone was created to deliver some of the envisioned marketing and value-added services.[2]

In March 2001, Microsoft announced its Passport system, which it intended to be the single buyer community for the entire Internet. Passport provides single sign-on for its affiliated Web sites such as the Microsoft Network (MSN) and Hotmail, and also provides a server-side wallet for storage of credit card numbers and other personal information such as bill-to and ship-to addresses. Some of Microsoft's competitors, led by Sun Microsystems, then announced the Liberty Alliance, whose purpose was to create an open, federated single sign-on system, to be followed by some kind of FCS. Microsoft then announced the opening up of Passport to a federated model. At the

2. CommerceTone was later acquired by Mercantec.

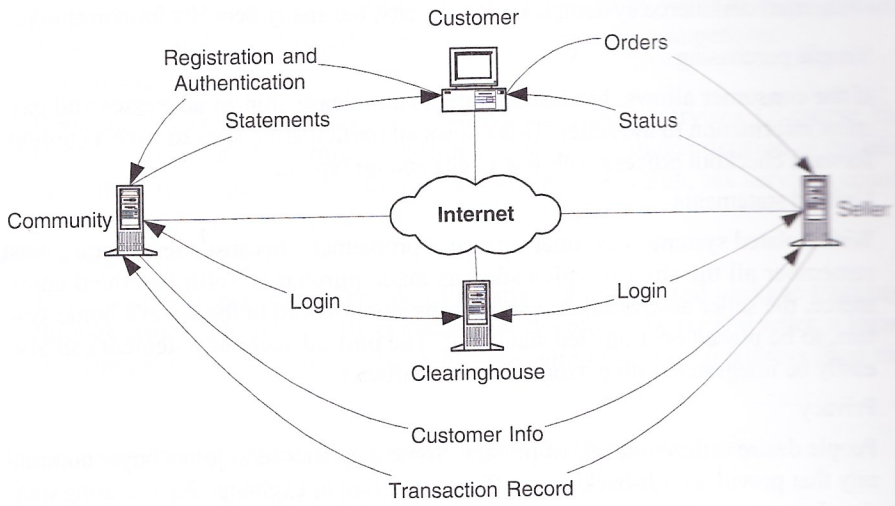


FIGURE 21-2. Federated Commerce System Transaction

time this book is being written, all of these systems are in the early stages of deployment, at best.

Let's follow a typical consumer transaction through its life cycle, with reference to Figure 21-2.

1. Previously, customer Jane joined a buyer community, completing its registration process.
2. Jane connects to the catalog Web site of the seller and selects items for purchase.
3. On the checkout screen, Jane selects **Use my FCS account**.
4. The seller system uses the services of the FCS clearinghouse to locate Jane's home community system (FCS Login).
5. Jane authenticates herself to her own home system.
6. The seller system requests buyer information from the home system, which Jane chooses to release.
7. The seller system completes the transaction and transmits a transaction record to Jane's unified customer statement held on her home system. This could be just a link to the seller's customer service module.
8. Later, Jane can view a unified list of all her FCS transactions and query any of them for status. The status link will take her to the particular seller system for current order status.

Consumer Services

The federated commerce system, if built correctly, has many benefits for consumers.

- Simple purchasing

  If the consumer allows, her home system can release ship-to addresses and payment information to the seller. This allows all participating sites to have a straightforward checkout process with few clicks and no typing.

- Unified statements

  With isolated systems, customer service is problematic because the customer must remember all the sites at which she has made purchases. With federated commerce, the seller always sends a short transaction record to the buyer's home system, to be placed on a unified statement. The unified account statement can also easily be integrated with personal finance software.

- Privacy

  People desire different levels of privacy. Some may choose to join a buyer community that provides cash-back or an affinity program in exchange for revealing additional personal information and accepting advertising. Others may choose to join a high-privacy community, which could go so far as to conceal completely the buyer's true identity from all other sites. In any event, the buyer has the ability to configure her personal privacy policy without the full-time job of studying the policies of every site visited.

- Multiple accounts

  Because there is a single point of control for an account, it is possible to put controls on the use of that account. This might be useful, for example, to allow children to shop online at selected sellers and with preestablished limits on their spending. The parents would have control over the limits on the account, and they would have access to a combined statement summarizing all activity on the family's accounts.

- Multistore gift registry

  Gift registries are typically limited to a single store or chain of stores. In a federated system, it would be straightforward to create gift registries that span multiple stores, providing more choices and easier shopping.

Value-Added Services

Federated commerce, by virtue of standardized APIs and service agreements, makes it straightforward to create value-added services, such as the following.

- Affiliate programs

  The FCS is a natural venue for affiliate programs. Rather than requiring affiliates to join seller programs individually, the affiliate can set up one account and potentially join many programs at once.

APPL-1008
APPLE INC. / Page 7 of 20

- Payment providers

  Payment providers can more easily connect with sellers by virtue of FCS standard interfaces. Standardized service agreements also promote competition.

- Tax calculation

  Domestic and international tax services can easily provide services. Of course, basic Web services do not require a federated system, but standardization of the relevant interfaces makes integration easier. For greater value, tax services could provide filing and payment as well.

- Logistics

  Logistics companies can offer rate calculators, shipping, and package tracking through standard interfaces. By connecting with both the buyer and seller, a logistics company could enhance privacy by concealing the ship-to address from the seller.

The FCS can offer many services, but they all build on three mechanisms.

1. The power of standardized APIs

   The FCS uses a Web services model for communications and integration. This means that interfaces are defined by agreed-on XML data types, and the existence and types of interfaces are defined by XML metadata. The FCS sets standards for the APIs to the network, which helps to ensure that systems provided by different vendors will interoperate.

2. The power of standardized contracts and directories

   Participants in the FCS can take advantage of standard roles. For example, someone can register as an affiliate and agree to the standard affiliate "deal" and then consult listings of sellers who offer that deal and sign up with many of them at once. Similarly, customers of a service who accept a standard deal can select suppliers who offer it. Standardized specifications of agreements such as percentage discounts for quick payment can even be processed automatically.

3. The power of networking

   Metcalfe's Law says that the value of a network increases as the square of the number of participants. Something similar seems to apply to the FCS: the aggregate expense of the network grows linearly with its size, but the savings grow with the number of possible interactions. In addition, the FCS is a boon for service providers, such as logistics companies, that must communicate with both the buyer and the seller. In standalone systems, this contact point is internal and inaccessible, but in the FCS, the contact between buyer and seller is external and easy to hook into.

Who Pays Whom?

As we have learned from the late 1990s, there are many possible business models for Internet services. It is probably easiest to compare parts of the FCS to the credit card system. In the credit card system, whenever a consumer uses a card, most of the

money goes to the merchant, but a portion of the payment is split among the merchant's bank, the cardholder's bank, and processing companies. Generally, small fixed fees go to those parties whose role is merely to route transactions, whereas larger, although still small, percentage amounts go to parties who take on the financial risks of fraud, default, and so on. This seems to be a good model for the FCS. Roughly speaking, the network itself, the *clearinghouse*, may qualify for a fixed processing fee, and the buyer's community may qualify for a fixed fee for record keeping, but unless parties take on financial risk, they should not qualify for a fee based on the size of the transaction.

System Functionality

In Chapter 2, we introduced the commerce value chain: attract and keep customers, turn interest into orders, manage orders, and provide customer service. Excellence in managing orders and providing customer service can help attract and keep customers. The more one learns about one's customers (with great attention to privacy), the better one can market to them.

Our implementation approach divides the components of the value chain into the following groups.

- Content servers and applications
- Linking the content to transactions
- Order management, including order processing, payment, fulfillment, and customer service

Walking Through a Transaction

Before we discuss the system architecture and design issues, we'll walk through a transaction.

Before opening for business, a seller creates an online catalog containing the product descriptions, prices, and so forth. The seller also creates a merchant profile, which describes the accepted means of payment and tax and shipping rules, and configures the fulfillment system to route orders.

When the buyer browses the catalog, the purchasing process begins, consisting of the following steps.

1. The buyer locates products of interest in the catalog and clicks the associated offers. This creates items in the buyer's shopping cart.
2. On checkout, the buyer sees the order form, an example of which is shown in Figure 21-3. The screen shown already has some buyer information filled in, as would be the case for a repeat buyer, based on saved member information.

3. The transaction engine calculates sales tax using the buyer's billing address, the seller's tax profile, and the product tax classification code.
4. The transaction engine calculates shipping charges using the buyer's choice of seller-defined shipping methods, together with product price or weight.
5. The buyer clicks **Buy Now**.
6. The transaction engine performs a credit card authorization and address verification check with the seller's choice of card processing network.
7. The transaction is recorded in permanent storage.
8. An advice of order received is routed to the seller.
9. The transaction engine returns a receipt to the customer, which contains a copy of the invoice and a link to a status page for customer service.
10. The seller ships the product and informs the transaction engine of the shipment, using the online order status pages.
11. The transaction engine performs credit card settlement, transferring funds from the buyer to the seller's account.
12. The transaction engine updates the buyer's online statement and optionally sends the buyer an e-mail advising of the order shipment.

Afterward, the buyer's receipt remains active, bringing the buyer to his online statement of activity. Each item is a hypertext link to a status page for the particular transaction.

Digital Goods

If the product purchased is in electronic form, the buyer's receipt is a hypertext link directly to the fulfillment area. Either the receipt carries with it security codes granting access by the buyer to specified areas within the fulfillment site for a specified period of time, or access control for the fulfillment area must be updated.

Subscriptions

If the product purchased is an ongoing subscription to an electronic publication, the transaction engine makes appropriate entries in its subscription database for the purposes of billing and control of access to the fulfillment servers.

When the buyer links to the fulfillment server at a later time, he will be challenged to authenticate himself against the customer database. Then an access control check will be performed against the subscription database. If the subscription is current, the buyer will be granted a ticket good for access to the content areas. Only when the ticket expires, perhaps weekly, will the buyer be required to reauthenticate himself.

APPL-1008
APPLE INC. / Page 10 of 20

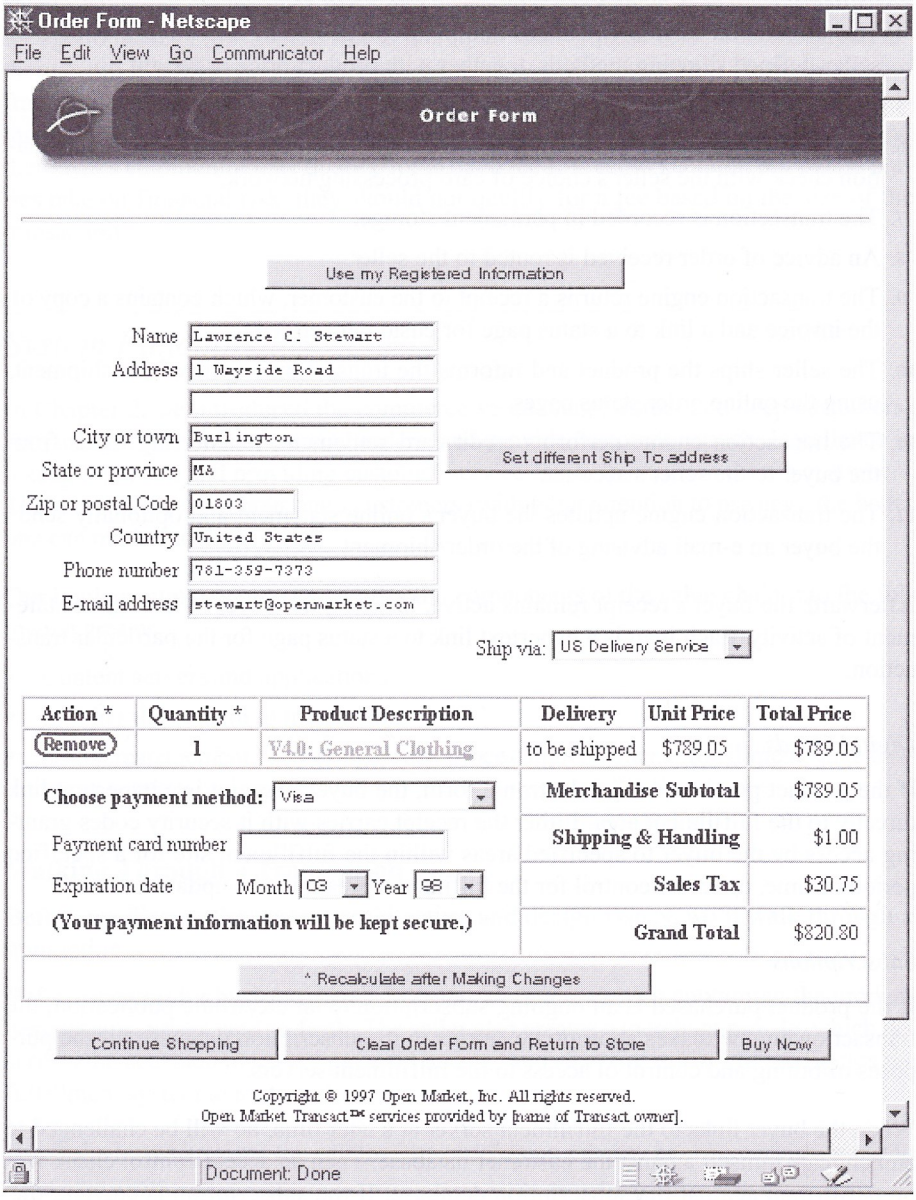


FIGURE 21-3. Example of Order Form

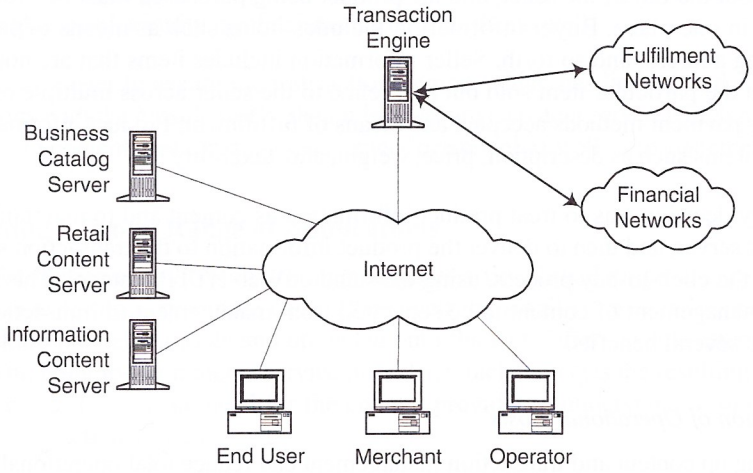


FIGURE 21-4. **Commerce Architecture**

System Architecture

When we introduced various architectures in Chapter 6, we said, "The architecture of a system defines its basic components and important concepts and describes the relationships among them." Figure 21-4 shows a block diagram of a commerce system built from these components. Overall, this is a three-tier architecture with clients using Web browsers connecting over the Internet to content servers of various types. The content servers in turn use shared commerce services. The application is interconnected over the public Internet. Finally, the transaction engine connects to services off the public Internet, such as financial networks and fulfillment systems.

The key architectural ideas are to separate management of content from management of transactions, to support a broad range of applications, to enable scaling of the system, and to accommodate evolution in technology and functionality.

Separating Content from Transactions

When we started thinking through the problems of commerce on the Internet, our first observation was that although we usually think of Web content as being present on a server somewhere on the network, it becomes displayable, rendered text and graphics only on the screen of the end user's Web browser. Consequently, the buyer's decision to buy will be communicated to the rest of the system as a network event, and we do not have to send that event to the same system that generated the page of content.

Our next observation was on the nature of a business transaction. Typically, information about the buyer, the seller, and the product being purchased must be brought together in one place. Buyer information includes items such as means of payment, shipping address, and so forth. Seller information includes items that are not dependent on the particular item sold but that relate to the seller across multiple products, such as payment methods accepted and means of fulfillment. Product information includes items such as description, price, weight, and taxability.

Our key decision was to treat product information as content and to maintain it on a content server, and then to deliver the product information to the transaction server as part of the click-to-buy process, using the standard Web HTTP protocol. This idea enables management of content to be separated from management of transactions, and permits several benefits.

Reduction of Operational Costs

Splitting up content and transaction management can reduce total operational costs.

Content servers have very different operational requirements than transaction servers. Indeed, if content is delivered by e-mail or CD-ROM, there may not be a content server at all.

Placing the transaction and business records in one place minimizes the number of systems that require database management systems and expensive reliability and integrity protections, and minimizes the number of systems that require specialized management and operations skills.

Because many content servers can share the services of a transaction engine, the costs of the transaction engine are amortized over many users. The content server costs are reduced because there is no need for each system to build in all the common costs of transaction handling.

Enabling of Service Providers

Multiple content servers can rely on shared commerce services provided by a transaction engine. This creates a new business—a commerce service provider, which operates a shared transaction system on behalf of multiple content servers. The transaction server can be shared across multiple divisions of a corporate enterprise or as a business in its own right.

Provision of Security Containment

A security systems is as strong as its weakest link. If valuable information is spread around the system, the security properties of the system must be equally strong everywhere. By placing mission-critical business functions and transaction records in a

shared transaction server, the security effort can be focused. The efforts of attackers may be concentrated on the transaction system "because that's where the money is," but the vigilance of security mechanisms is concentrated as well.

It is actually easier to secure the transaction engine, which has limited and stylized connections with the outside world, than it is to secure a content server, which must be accessible to content creation and management personnel as well as to customers.

Supporting a Broad Range of Applications

A shared commerce services environment enables a very broad range of commerce applications. At the very simple end of the spectrum, small stores can be prepared with desktop publishing tools and uploaded onto the Net. Offers to sell, encoded in the content, point to a commerce service provider, which manages the resulting orders on a fee basis. There is no need for the content provider to understand or operate a complex transaction system.

In the midrange, dynamic content and rich site-development tools can create a compelling commerce experience for the consumer and provide compelling value for the business user.

At the high end, an enterprise may have multiple divisions with different content systems, sharing transaction services provided by the IT department. The transaction engine may be integrated closely with enterprise financial and logistics systems.

By leveraging a shared infrastructure, small and medium-sized enterprises can access all the services that are available to large enterprises.

Supporting System Scaling

Splitting up content and transaction systems allows them to scale independently. Content systems need to grow according to the type and volume of content, which may be unrelated to the volume of transactions. A dynamic content site delivering pages rich in graphics and multimedia will have greater computing and bandwidth requirements than a content system delivering even a large number of simple pages. On the fulfillment side for digital goods, a single transaction may grant very substantial access rights to fulfillment servers over an extended period.

With the shared services design, content servers can be added to the system as the need for them arises as a result of increased demand for or volume of content. Content servers representing additional businesses may register for commerce services at any time.

The transaction engine also needs to be scalable. We designed it as a set of logical servers backed by a relational database management system. The application logic scales by running the component servers on more powerful hardware or by load shar-

ing. The database scales by running on more powerful hardware and by exploiting the database vendors' parallel and cluster capabilities.

Supporting System Evolution

An architecture is needed when one doesn't know what kinds of problems will arise tomorrow. Internet commerce is changing very rapidly, and so systems installed today will need to add new functionality and exploit new technologies as the industry matures.

Functional Evolution

Separating content and transaction systems permits functions to be added to each system without affecting or requiring upgrades to the other. For example, a new payment method such as purchase orders can be added in one place at the transaction server, making the new functionality immediately available to all affiliated content servers.

A content server employing dynamically generated pages can replace a server using static pages without any change in the transaction machinery.

Technological Evolution

An important part of an architecture is defining interfaces between the parts of the system that remain stable across multiple releases and technological changes. For example, server-side wallets may replace HTML forms and SSL/TLS for credit card payment on the Internet. When wallet support is added to the transaction engine, all the affiliated content servers can take immediate advantage of it without having to upgrade each content system—perhaps using several different content creation and management systems.

Transaction Engine

The best way to think of the transaction engine is as a suite of applications for its different users: buyers, sellers, and operators. This model is shown in Figure 21-5. In more traditional business settings, buyers interact directly with sellers, and both interact directly with operations and support groups. On the Internet, the situation is different because the parties involved are rarely online at the same time. Instead, each group of users interacts with applications that place their retained state in a database where the next group can pick it up and carry on. The database, without loss of generality, can be a set of databases with workflow capabilities.

The transaction engine is organized in this way, with application logic and user interface capabilities for buyers, sellers (merchants), and system operators. It is a rich application; the following sections are not exhaustive but are included here to give

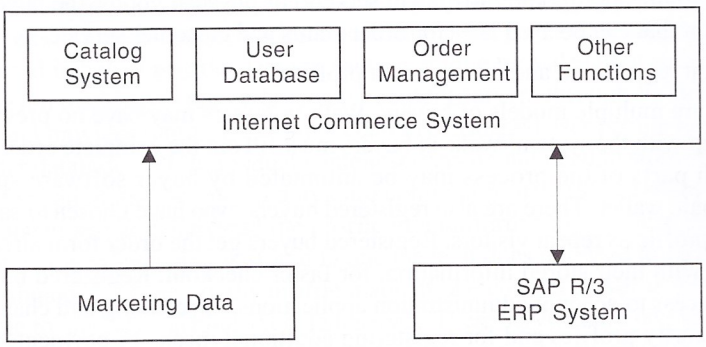


FIGURE 21-5. **Transaction Engine Application**

readers an appreciation of the kinds of functionalities and APIs that are included. In many implementations, the transaction engine includes the user interface, but in our example, the user interface for the buyers is specifically provided by the content management and catalog systems through transaction engine APIs.

Buyer Applications: User Interface and APIs

The buyer experience is primarily with catalog and other content systems. In the case of high-end content systems, or those with complex marketing and promotions logic, buyer interactions with the transaction engine are indirect, through content server user interfaces supported by application programmer interfaces provided by the transaction engine. In support of simpler catalog systems or of offers made by advertisements, buyers interact directly with transaction engine user interface software.

Buyer Applications with User Interface

- Order capture

  The order capture application appears to be an interactive order form. Buyers populate the order form directly by clicking digital offers and digital coupons, or the seller may prepopulate the order form through an API. The order form provides the facilities for the buyer to adjust item quantities, choose coupons, enter or edit billing and shipping addresses, and select payment and shipping methods. In addition to providing the order form user interface, the order capture subsystem calculates discounts from coupons (including matching coupons against items), validates addresses, and calculates sales tax and shipping charges, leading to an order total. If a consumer payment method such as a credit card is selected, the transaction engine obtains the necessary authorization in real time. The principle is to validate the success of the order to the fullest extent possible while the buyer is online, in order to resolve any problems right away. In the case of digital goods

and subscriptions, the buyer experience ends with a receipt, granting immediate access to the appropriate fulfillment server. For physical goods, the buyer obtains a receipt that can be used later for order status and customer service.

- Account registration and management profiles

  There are multiple models of buyers. Walk-in buyers may have no previous relationship with the system. Walk-in buyers must fill out the complete order form, although parts of the process may be automated by buyer software such as an electronic wallet. There are also registered buyers, who have chosen to save an account profile as repeat visitors. Registered buyers get the order form already populated with their stored information, for faster checkout. Registered buyers also have access to account administration applications, for creating and changing user and security profiles and for registering additional means of payment, such as a microtransaction account. Finally, there are buyers whose profiles are held elsewhere by a buyer community that is part of the FCS. These buyers can be handled either as walk-in buyers whose order forms are filled out automatically or as registered buyers, some of whose registration information is held remotely.

- Customer service

  The goal of the customer service applications is customer self-service. For registered buyers, the system provides online statements of all transaction activity. Each item is active and can be clicked through to reach detailed order status information or the catalog page from which the item was originally selected. From the order status screen, the buyer can click through to the package tracking services offered by some logistics companies. The subscription capabilities are also supported by some special customer service applications. For example, if a subscription involves periodic payments and the associated credit card expires or is canceled, the system permits the buyer to substitute a different payment instrument without intervention by the operator.

Buyer APIs

Buyers will not use these APIs themselves. Instead, these are APIs that sellers who have relationships with the transaction engine use in order to implement the buyer experience.

- Commerce objects

  The commerce objects, including offers, coupons, and receipts, constitute the primary API to the transaction engine. These objects are the primary mechanisms through which items for sale, merchandising, and fulfillment of digital goods are handled between content servers and the transaction engine.

- Order injection API

  When a buyer is online, the content server can implement the user interface for the complete buyer experience and then hand off a complete order. In effect, the user interface for the order capture phase of commerce can be remotely located at a content server.

- Customer database API

  As we discussed in Chapter 12, in order to deploy sites with personalized content, a site needs to identify users, track sessions, and build pages according to information stored in a user profile. The transaction engine provides user authentication and session tracking to remotely located content servers through its subscription server and provides remote access to a centrally managed user profile through the customer database API. Individual content servers do not need to store authentication information and do not need to store common profile information, although each content server can extend the profile in a uniform way. Because profile and authentication information is stored in one place, it does not need to be replicated or distributed. A change in the profile made at one content server is immediately available to any other affiliated content server. These services are provided by the SecureLink digital query. When a content server or catalog server participates in the FCS, the customer database API links the content server to whatever buyer community system is the buyer's home system.

Seller Applications and APIs

In order to run an Internet-based business, a seller needs to create content and to configure and manage a wide variety of business rules and services. In our model, catalog content or site content is prepared and managed with a variety of applications and tools, but commerce facilities are centrally managed.

Seller Applications

- Merchant profiles

  In order to run their businesses, the sellers or merchants must configure a variety of business rules. The transaction engine provides a merchant administration application organized as a series of profile screens. The seller configures which payment instruments (such as credit cards) will be accepted from among those offered and, for each payment instrument, how processing is to be handled. For sales tax or value-added tax, the seller provides information on tax jurisdictions and related business rules. For shipping of physical goods, the seller configures an order delivery mechanism and one or more shipper profiles, which provide the business rules for calculating shipping costs displayed to the buyer.

- Order management and fulfillment

  The transaction engine can deliver orders to the seller by online or downloaded report, by fax, by regular or secure e-mail, or by use of the fulfillment API discussed later. The seller has access to tax and transaction statements and reports and can mark orders as *fulfilled* and provide an appropriate e-mail message for the buyer.

APPL-1008
APPLE INC. / Page 18 of 20

- Customer service

  Inevitably, questions and problems arise in handling orders. Our objective is to let buyers resolve their own questions through online customer service but also to provide means for the seller to process returns and credits.

Seller APIs

Most of the seller-oriented APIs were discussed under Buyer APIs because they relate to buyer activities during browsing and purchasing online. The order injection API, however, can also be used in an offline manner.

- Order injection API

  Even when order capture takes place in a medium other than the Internet, there is a benefit in using an Internet commerce system for order management and customer service. The order injection API permits entry of orders originated offline and then applies standard processing and online customer service. This permits both buyers and sellers a unified view of orders.

Operator Applications and APIs

The transaction engine can be operated by a single enterprise on its own behalf, or it can be run by a commerce service provider as a business providing Internet commerce services for other companies. In the first instance the operator applications and APIs are really seller facilities, whereas in the latter instance the operator applications and APIs support central services and policies of the commerce service provider.

Operator Applications

- Configuration

  The transaction engine is designed for great flexibility of installation, business rules, and user interfaces. The operator is responsible for using standardWeb tools to create a standard look and feel for the system, even if the customization is only the addition of a name and a logo to the standard templates. The operator must also make a variety of policy decisions about trust, security, payment processing, and privacy. The operator is responsible for installing support for particular payment mechanisms and payment processing systems and configuring them for proper operation.
- Administration

  During normal operation, the transaction engine provides application support for ongoing activation and modification of affiliated content servers and central overrides for payment processing and merchant profiles.

- Reporting and logging

  The transaction engine creates log files for error reporting and audit trails and provides reporting tools for various kinds of system activities and for support of auxiliary activities such as fraud control.

Operator APIs

- Customer database API

  In addition to its use for real-time customer profile access from content servers, the customer database API permits individual or bulk loading of customer information. This makes it possible to Internet-enable existing systems.

- Fulfillment API

  In those cases in which the Internet order management application must directly connect with existing enterprise order management and fulfillment systems, the fulfillment API permits programmatic access to pending orders and dynamic updates of order status on a per-line-item basis.

- Payment API

  The transaction engine contains a variety of built-in Internet-capable payment systems, including credit cards, purchase orders, and microtransactions. However, Internet payment technology is changing rapidly, with debit cards, electronic funds transfer, and electronic checks gaining acceptance and with other, more innovative systems on the horizon. The payment API permits new payment systems and additional implementations of old ones to be added. Catalog content is linked to the transaction engine, which operates as a payment switch,[3] so new payment systems can be added without the necessity of updating any content servers.

System Functionality

The system is designed with groups of functionality for several Internet commerce market segments: business-to-business, business-to-consumer, and information commerce. Many if not most of the differences in required functionality for these markets are addressed by different content systems—catalogs for business-to-business commerce, catalogs and entertainment sites for business-to-consumer commerce, and complex search and retrieval engines for information commerce—but the business rules, payment methods, and common services are also distinct.

3. See, for example, "Payment Switches for Open Networks," by D. K. Gifford, L. C. Stewart, A. C. Payne, and G. W. Treese, IEEE COMPCON (March 1995).

APPL-1008
APPLE INC. / Page 20 of 20