

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών



Μεταπτυχιακό Πρόγραμμα
Τεχνο-οικονομικά Συστήματα

ΜΑΘΗΜΑ: Ηλεκτρονικές Συναλλαγές

ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΣΥΝΑΛΛΑΓΩΝ II

ΣΥΝΑΛΛΑΓΩΝ II

Διδάσκοντες:

Επικ. Καθηγητής Δ. Ασκούνης, Δρ. Ι. Χαραλαμπίδης

The Landscape and Framework of EC Applications Development

Step 1: Identifying, Justifying, and Planning EC Systems

Step 2: Creating an EC Architecture

EC architecture

A plan for organizing the underlying infrastructure and applications of a site

Step 3: Selecting a Development Option

The Landscape and Framework of EC Applications Development

- Step 4: Installing, Testing, Connecting, and Deploying EC Applications
 - Unit testing**
Testing application software modules one at a time
 - Integration testing**
Testing the combination of application modules acting in concert
 - Usability testing**
Testing the quality of the user's experience when interacting with a Web site
 - Acceptance testing**
Determining whether a Web site meets the original business objectives and vision
- Step 5: Operations, Maintenance, and Updating

The Landscape and Framework of EC Applications Development

- Managing the Development Process
 - For medium-to-large applications, a project team is usually created to manage the process and the vendors
 - Collaboration with business partners is also critical
 - Projects can be managed with project management software. Appropriate management also includes periodic evaluations of system performance
 - Implementing an EC project may require restructuring of one or more business processes

The Major EC Applications and Their Functionalities

- B2C Storefronts
 - Product presentation function
 - Order entry function
 - Electronic payment function
 - Order fulfillment function
 - Customer service function
 - Product support function

The Major EC Applications and Their Functionalities

- Supplier Sell-Side B2B Sites
 - Personalized catalogs and Web pages for all major buyers
 - A B2B payment gate
 - Electronic contract negotiation features
 - Product configuration by customers
 - Affiliate program capabilities
 - Business alerts (e.g., to special sales, to news)

The Major EC Applications and Their Functionalities

- E-Procurement
 - Aggregating Catalogs
 - Reverse Auctions and Tendering Systems
 - Forward Auctions
 - Exchanges
 - Portals
 - Other EC Systems

Development for EC Applications

- In-House Development: Insourcing

Insourcing

In-house development of applications

- Development Options
 - Build from scratch
 - Build from components
 - Enterprise application integration

Development for EC Applications

- Buy the Applications
 - This option is also known as a *turnkey approach*. It involves buying a commercial package, installing it, and starting it up
 - When selecting a particular package, the package should not only satisfy current needs, it must also be flexible enough to handle future ones; otherwise the package may quickly become obsolete
 - It is sometimes necessary to acquire multiple packages

Development for EC Applications

- Leasing
 - Lease the application from an *outsourcer* and then install it on the company's premises
 - Lease the application from an *application service provider (ASP)* that hosts the application at its data center
 - A variation of the preceding option is to use utility computing, described later in this chapter, which is described in detail later on in this chapter

Development for EC Applications

- Outsourcing and Application Service Providers (ASP)
 - Outsourcing

Application Service Provider (ASP)

A company that provides business applications to users for a small monthly fee

Development for EC Applications

Utility Computing

Unlimited computing power and storage capacity that can be used and reallocated for any application—and billed on a pay-per-use basis

- Policy-based service-level-management tools
- Policy-based resource-management tools
- Virtualization tools

Exhibit 18.4 The Five Elements of a Successful Utility-Computing Value Proposition



Development for EC Applications

- Other Development Options
 - Join an e-marketplace
 - Join an auction or reverse auction third-party site
 - Joint ventures
 - Join a consortium
 - Hybrid approach

Criteria for Selecting a Development Approach

- Flexibility
- Information requirements
- User friendliness
- Hardware and software resources
- Installation
- Maintenance services
- Vendor quality and track record
- Estimating costs

Criteria for Selecting a Development Approach

- Personnel
- Technological evolution
- Scaling
 - Scalability**

How big a system can grow in various dimensions to provide more service; measured by total number of users, number of simultaneous users, or transaction volume
- Sizing

Criteria for Selecting a Development Approach

- Performance

 - **Latency**

 - The time required to complete an operation such as downloading a Web page.

 - **Throughput**

 - The number of operations completed in a given period of time; indicates the number of users that a system can handle

- Reliability

- Security

Third-Party Providers of EC Software and Suites

- Functional Software Packages

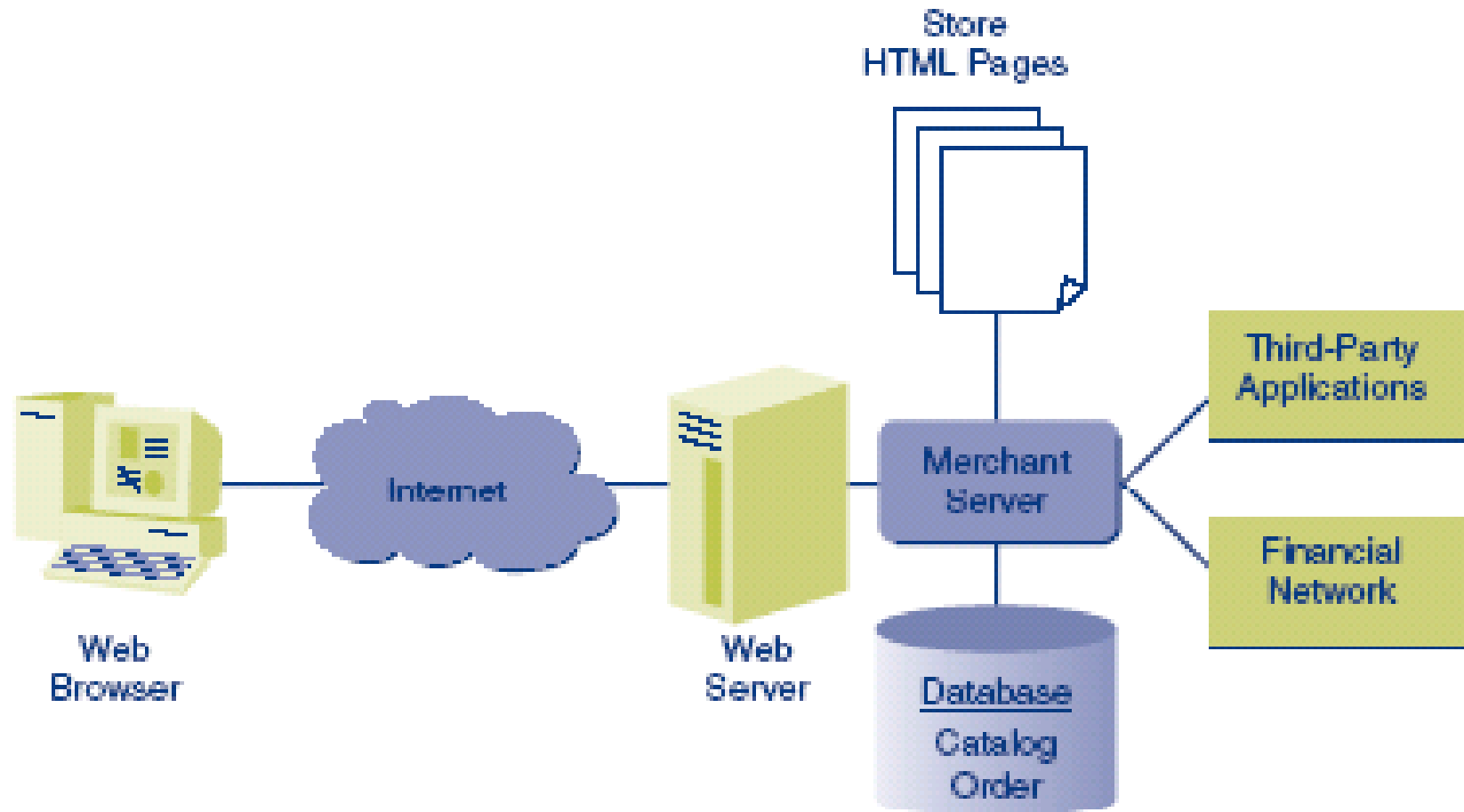
- **Electronic catalog**

- The virtual-world equivalent of a traditional product catalog; contains product descriptions and photos, along with information about various promotions, discounts, payment methods, and methods of delivery.

- **Merchant server software**

- Software for selling over the Internet that enables companies to establish selling sites relatively easily and inexpensively

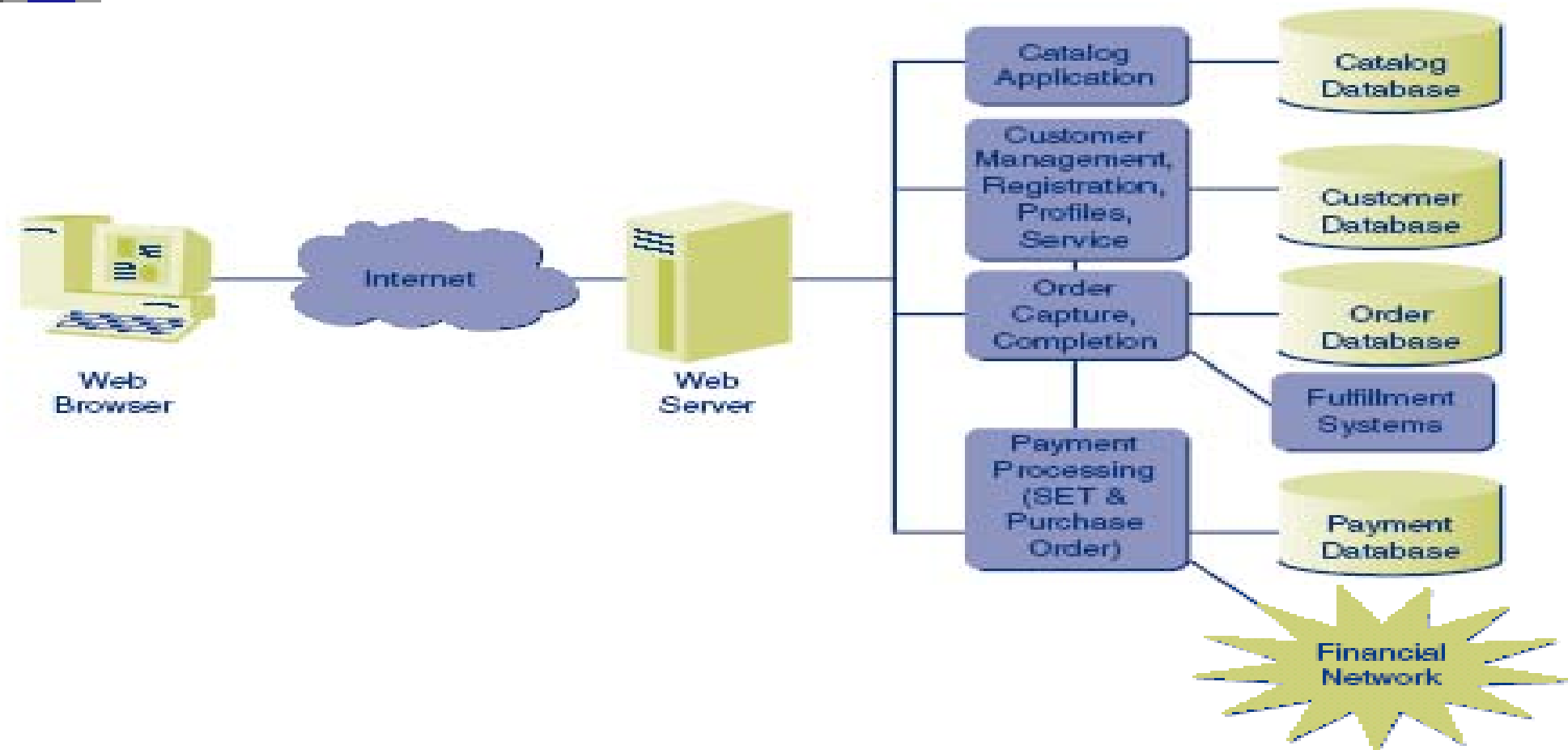
Exhibit 18.5 Merchant Server Architecture



Third-Party Providers of EC Software and Suites

- EC Suites
 - Microsoft's Commerce Server
 - IBM's WebSphere Commerce Suite
 - Oracle's EC Products
 - Other EC Suites

Exhibit 18.6 Major Components of an EC Suite



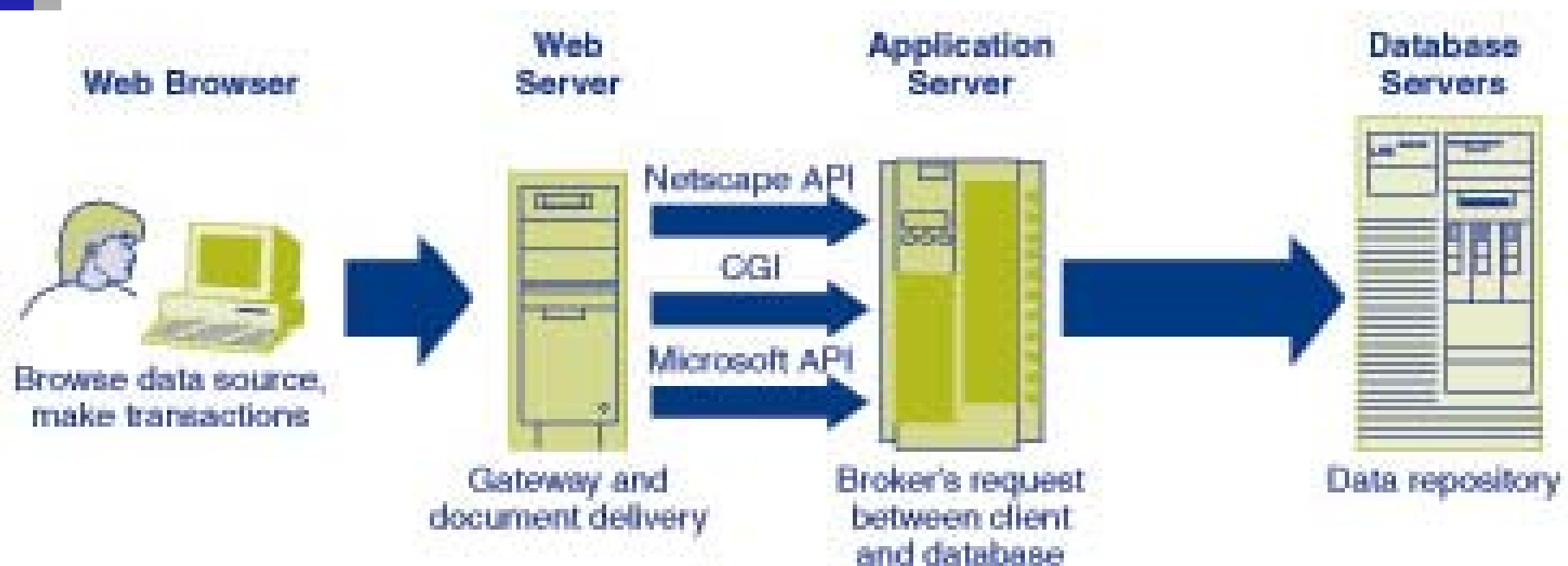
Connecting to Databases and Other Enterprise Systems

- Connecting to Databases

Multitiered application architecture

EC architecture consisting of four tiers: Web browsers, Web servers, application servers, and database servers

Exhibit 18.7 Example of Multitiered Application Architecture Connected to Database



Connecting to Databases and Other Enterprise Systems

- Integrating EC Applications and Back-End Systems

Enterprise application integration (EAI)

Class of software that integrates large systems

Connecting to Databases and Other Enterprise Systems

- **Middleware**
 - Hiding distribution (i.e. the fact that an application is usually made up of many interconnected parts running in distributed locations)
 - Hiding the heterogeneity of the various hardware components, operating systems, and communication protocols
 - Providing uniform, standard, high-level interfaces to the application developers and integrators, so that applications can be easily composed, reused, ported, and made to interoperate
 - Supplying a set of common services to perform various general purpose functions, to avoid duplicating efforts and to facilitate collaboration between applications

Rise of Web Services, SML, and Service-Oriented Architecture

- The Need for Integration
 - Platform-specific objects
 - Dynamic environment
 - Security barriers

Rise of Web Services, SML, and Service-Oriented Architecture

- The Roles of XML and Web Services

Web service

A software system identified by a URI (uniform resource indicator), whose public interfaces and bindings are defined and described using XML

Rise of Web Services, SML, and Service-Oriented Architecture

- Web Services

Web Services are self-contained, self-describing business and consumer modular applications, delivered over the Internet, that users can select and combine through almost any device, ranging from personal computers to mobile phones

Rise of Web Services, SML, and Service-Oriented Architecture

- Key Technologies in Web Services

- **Simple Object Access Protocol (SOAP)**

- Protocol or message framework for exchanging XML data across the Internet

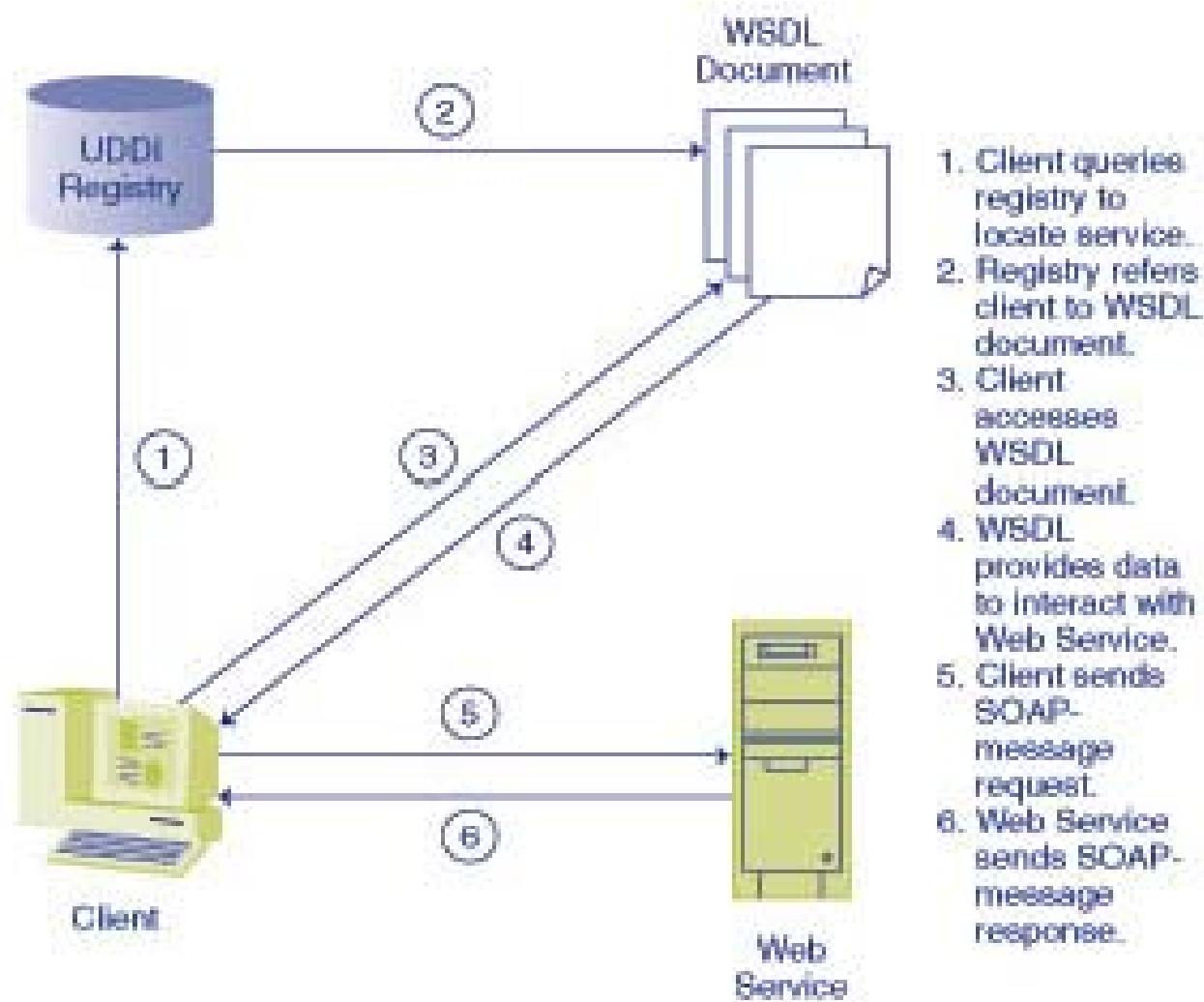
- **Web Services Description Language (WSDL)**

- An XML document that defines the programmatic interface—operations, methods, and parameters—for Web Services

Rise of Web Services, SML, and Service-Oriented Architecture

- Key Technologies in Web Services
 - universal description, discovery, and integration (UDDI)**
An XML framework for businesses to publish and find Web Services online
 - security protocol**
A communications protocol that encrypts and decrypts a message for online transmission; security protocols generally provide authentication

Exhibit 18.8 Key Components of Web Services



Rise of Web Services, SML, and Service-Oriented Architecture

- Web Services Platforms
 - Microsoft .NET
 - IBM WebSphere
 - J2EE Architecture

Rise of Web Services, SML, and Service-Oriented Architecture

- The Notion of Web Services as Components
 - Web Services enable the Web to become a platform for applying business services as components in IT applications
 - Web Services offer a fresh approach to integration
 - Business processes that are comprised of Web Services are much easier to adapt to changing customer needs and business climates than are “home-grown” or purchased applications

Rise of Web Services, SML, and Service-Oriented Architecture

- A Web Services Example
 - Consider an airline Web site that provides consumers with the opportunity to purchase tickets online
 - The airline recognizes that customers also might want to rent a car and reserve a hotel as part of their travel plans
 - The consumer would like the convenience of logging onto only one system rather than three, saving time and effort
 - Also, the same consumer would like to input personal information only once

Rise of Web Services, SML, and Service-Oriented Architecture

- Web Services Entering the Mainstream
 - Google's Web Services API (application programming interface) enables programmers and application developers to issue SOAP-based search requests to Google's index of more than 3 billion Web pages and to receive results as XML data
 - Amazon.com also offers an extensive set of Web Services that can be used by its "Associates" and other product sellers and vendors

Rise of Web Services, SML, and Service-Oriented Architecture

- Advantages and Hurdles in Implementing Web Services

Over the years, a number of programming initiatives have attempted to solve the problem of interoperability (i.e., getting software and applications from different vendors running on different hardware and operating systems to communicate with one another in a transparent fashion).

Rise of Web Services, SML, and Service-Oriented Architecture

- Advantages of Web Services
 - Rely on universal, open, text-based standards that greatly simplify the problems posed by interoperability and that lower the IT costs
 - Enable software running on different platforms to communicate, reducing the cost and headaches of multiple platforms
 - Promote modular programming, which enables reuse by multiple organizations

Rise of Web Services, SML, and Service-Oriented Architecture

- Advantages of Web Services
 - Are easy and inexpensive to implement because they operate on the existing Internet infrastructure. They also offer a way to maintain and integrate legacy IT systems at a lower cost than typical EAI efforts
 - Web Services can be implemented incrementally

Rise of Web Services, SML, and Service-Oriented Architecture

- Barriers to Adoption of Web Services
 - The standards underlying Web Services are still being defined, thus interoperability is not automatic
 - Web Services standards are not well-defined is security
 - Although Web Services rely on XML as the mechanism for encoding data, higher-level standards are still required, especially in B2B applications

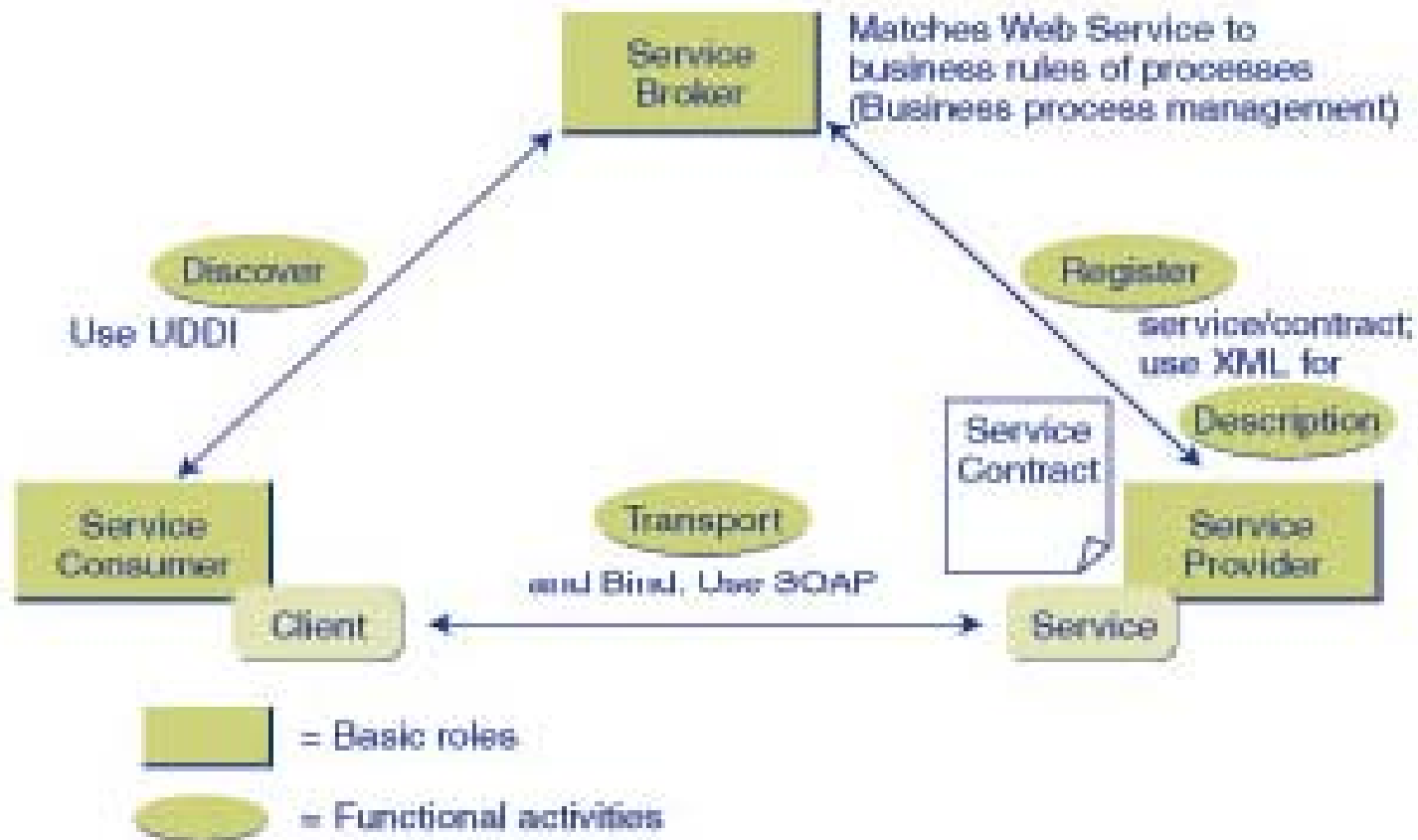
Rise of Web Services, SML, and Service-Oriented Architecture

Service-Oriented Architecture (SOA)

An application architecture in which executable components, such as Web Services, can be invoked and executed by client programs based on business rules

- Transport
- Description
- Discovery
- Registration

Exhibit 18.9 The Three Conceptual Roles and Operations of a Service-Oriented Architecture



Vendor and Software Selection

Step 1: Identify Potential Vendors

Step 2: Determine the Evaluation Criteria

Request for proposal (RFP)

Notice sent to potential vendors inviting them to submit a proposal describing their software package and how it would meet the company's needs.

Step 3: Evaluate Vendors and Packages

Vendor and Software Selection

Step 4: Choose the Vendor and Package

Step 5: Negotiate a Contract

Step 6: Establish a Service Level Agreement

Service level agreement (SLA)

A formal agreement regarding the division of work between a company and a vendor.

Usage Analysis and Site Management

- Log Files

- **Access log**

- A record kept by a Web server that shows when a user accesses the server; kept in a common log file format, each line of this text file details an individual access

- Pageviews by time slot
 - Pageviews by customers' log-in status
 - Pageviews by referrers
 - Pageviews by visitor's hardware platform, operating system, browser, and/or browser version
 - Pageviews by visitor's host

Usage Analysis and Site Management

- E-Commerce Management Tools
 - Several vendors offer suites of products or individual packages that can assist with the management process
 - Other EC management tools include site version control tools, combined utilities/tools, server management and optimization tools, and authoring/publishing/deployment tools that include significant site management or testing capabilities

Managerial Issues

1. What is our business perspective?
2. Do we have a systematic development plan?
3. Insource or outsource?
4. How should Web Services be deployed?
5. How should we choose a vendor/software?
6. Have we analyzed the data?
7. Should users be involved?
8. How should we manage development risks?
9. How shall we plan for service-oriented architecture
10. (SOA)?

Summary

1. The major steps in developing an EC application.
2. The major EC applications and their major functionalities.
3. The major EC application development options along with their benefits and limitations.
4. EC application outsourcing options.
5. The major components of software packages and EC application suites.
6. Methods for connecting an EC application to backend systems and databases.

Summary

7. The rise of Web Services and XML.
8. Understand service-oriented architecture and its relationship to EC.
9. Criteria used in selecting an outsourcing vendor and package.
10. The value and uses of EC application log files.
11. The importance of usage analysis and site management.