

# 3. Enabling Web and Software Technologies

---

**3.1 Client / Server Architectures and the Internet**

**3.2 Internet Protocols and Internet Infrastructure**

**3.3 Multi-Tier Architectures**

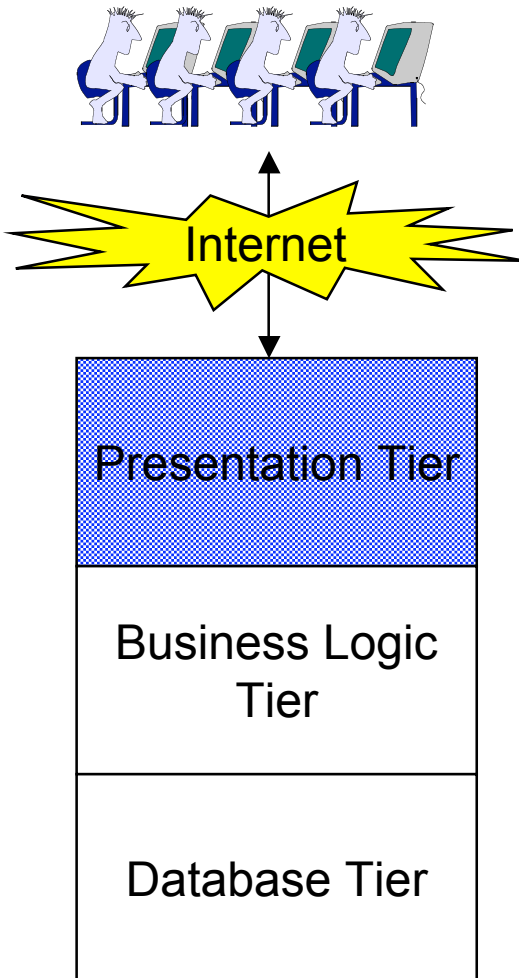
Three-tier Architectures

Enterprise Architectures: J2EE Architecture, .NET Architecture

**3.4 Platform Choices and Connectivity Options**

# Three-Tier Architectures: Presentation Tier

---



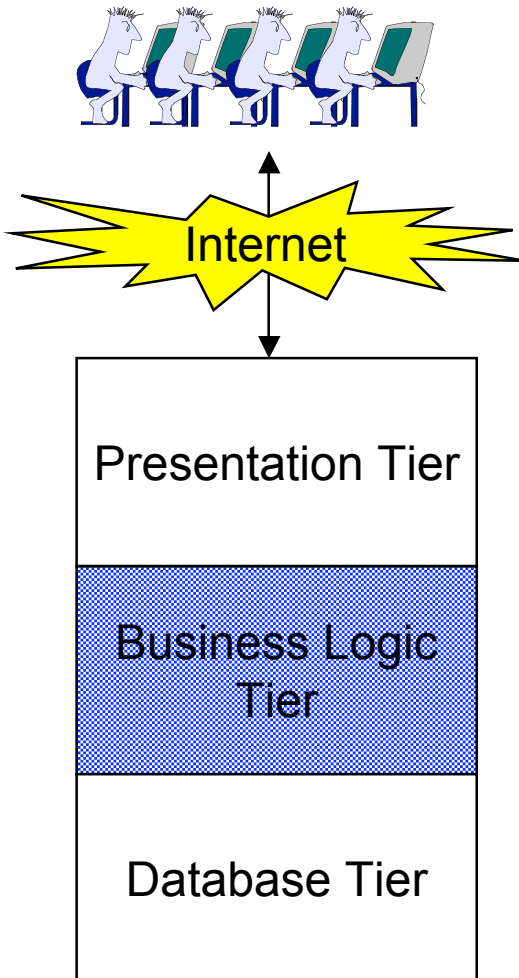
Presentation Tier:

Purpose: Content Visualizations (static and dynamic HTML pages, WML pages).

Sample Realization Technologies:

- Web Server for static content,
- Engines for dynamic content.

# Three-Tier Architectures: Business Logic Tier



Business Logic Tier:

Purpose: Modeling of Business Processes and Business / Domain Objects

Refined View:

- Business Processes*
- Containers* managing the Business Objects' life-cycle and provide search functionality
- Business Objects*

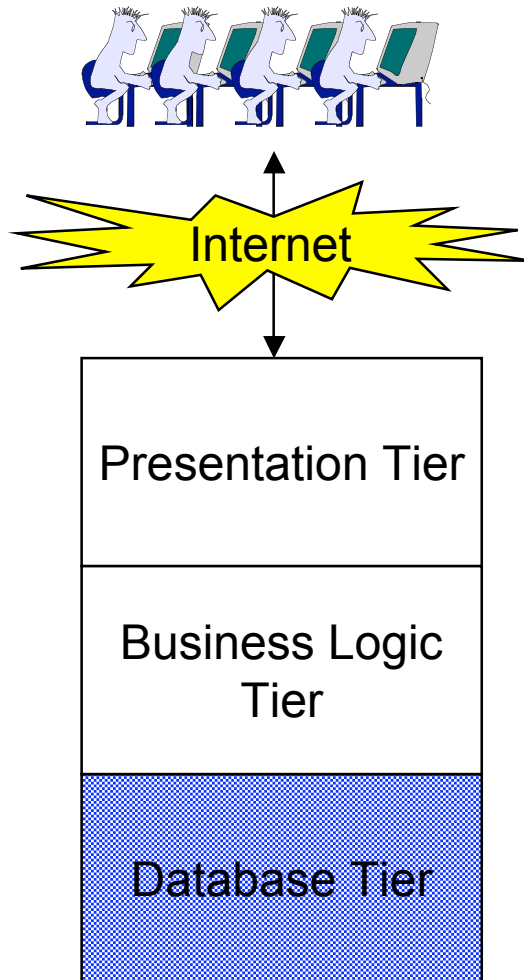
Business Processes define the interface to the users: „What can a user do in an ECommerce system (shop, knowledge portal, expert site)?“

They

- use the containers to
  - create new business objects (empty shopping cart)
  - delete unused business objects (sold products) and
  - find business objects (product #4711, a user)
- manipulate business objects (ex: change user state to „logged in“).

# Three-Tier Architectures: Database Tier

---



Database Tier:

Purpose: Persistence, Failure recovery for Business Objects, Business Process State and more.

Sample Realization Technologies:

- ❑ Java: RDBMS and JDBC (Java Database Connectivity)

# 3. Enabling Web and Software Technologies

---

**3.1 Client / Server Architectures and the Internet**

**3.2 Internet Protocols and Internet Infrastructure**

**3.3 Multi-Tier Architectures**

Three-tier Architectures

Enterprise Architectures: J2EE Architecture, .NET Architecture

**3.4 Platform Choices and Connectivity Options**

# Enterprise Architectures

---

## Current Enterprise (Server-Side) Component Architecture:

J2EE / Enterprise JavaBeans (EJB): Java-based component architecture

+ Platform neutral

- Language dependent (Java). Requires components to be written in the Java language (adding components using CORBA is not yet practical)

Microsoft .NET: Microsoft-platform component architecture (following DCOM)

+ Language independent: Components can be written in any .NET supported language. Example: C#, J#, VisualBasic.NET, C++,

+ Components are developed using existing IDE (VisualStudio.NET)

- Platform dependent: Bound to Microsoft Windows as platform (officially called platform-neutral)

*You can learn more about Component Architectures in the **Software Architectures, Architectures for Enterprise Applications, and Web Engineering lectures.***

# J2EE Architecture

---

The **Java 2 Enterprise Edition (J2EE)** platform defines a Java-based architecture for (web-based) Enterprise Software Systems.

The J2EE architecture follows the three-tier component architecture, i.e., it has a logical software partitioning into the layers:

- ❑ The **presentation layer** contains components dealing with user interface and user interaction.

Technologies: Java Server Pages (JSP), Java Servlets and/or Java Applets.

- ❑ A **business logic layer** contains business process and business entity components that work together to solve business problems.

Technologies: Enterprise JavaBeans.

- ❑ A **data layer** is used by the business logic layer for persistence of entity states and - optionally - process states. Central to the data layer are one or more databases that house the persistent states [Roman99].

Technologies: RDBMS (Oracle, MS SQL Server, MySQL, ...)

# J2EE Architecture (2)

---

Presentation layer



Business logic layer



Storage layer



# Enterprise JavaBeans (EJB)

---

**Enterprise JavaBeans (EJB)** is a *server-side component architecture* that enables and simplifies the process of building enterprise-classes for distributed object applications in *Java*. By using EJB, you can write scalable, reliable, and secure applications without writing your own complex distributed object framework [Roman99].

Enterprise JavaBeans are a part of the J2EE architecture. The Enterprise JavaBeans are components realizing the business logic layer. Enterprise JavaBean components can be

- ❑ non-persistent objects (Session Beans) or
- ❑ persistent objects (Entity Beans)

Enterprise JavaBeans are deployed to an **EJB Server (Application Server)**, in which they are running inside an **EJB Container** that realizes

- ❑ **Persistence**
- ❑ **Transactions**
- ❑ **Transparent Distribution**
- ❑ **Concurrency Support**
- ❑ **Recovery / Transparent fail-over for entity beans - and optionally for session beans**
- ❑ The components (EJBs) do not have to provide these functionalities.

# EJB Specification(s)

---

The ***EJB specification*** defines roles and components and describes how components should work together. The specification is implemented by several EJB server vendors.

Further reading:

- ❑ [www.java.sun.com/products/ejb/index.html](http://www.java.sun.com/products/ejb/index.html) (EJB home page)
- ❑ [www.theserverside.com](http://www.theserverside.com) (EJB spec 2.0 book by Ed Roman available online)
- ❑ An EJB vendor comparison can be found at <http://www.flashline.com/Components/appservermatrix.jsp>

# EJB: Choice of Application Server

---

## Criteria for EJB Servers:

- Compliance to J2EE requirements
- Proven real-world deployments
- Suitability for application to be deployed (e.g. if the application just consists of JSPs / Servlets and is not expected to include EJBs or Java Messaging Services (JMS))
- Stability
- Integration with third-party tools
- Tools supporting the deployment process
- Administration: Ease-of-maintenance and usability
- After-sales support

**Evaluation of EJB servers:** Evaluation of application servers, since almost all leading EJB server products can be downloaded / requested for an limited-time evaluation

## Important:

Evaluation helps find performance patterns of the servers for the given application domain, as well as make evident whether the performance comes “built-in” in the server, or is a result of server-specific enhancements, which would create problems with portability of the application.

[Kaleem00]

# J2EE / EJB Reference Implementation by Sun

---

Sun provides a reference implementation of the J2EE platform and the EJB specification:

- Downloadable from <http://www.java.sun.com/j2ee/>
- Not very performative but good for a start
- Includes Cloudscape database
- Easy to configure (Configuration tool included)
- Usable for sample (toy) systems
- Free

Other EJB Servers are:

- BEA: WebLogic Server
- IBM: WebSphere
- jBoss: jBoss Server (Open Source project)
- ...

# Microsoft .NET Architecture

---

The .NET architecture comprises:

Presentation Layer:

- ❑ Microsoft Active Server Pages Plus (ASP+), comparable to Java Server Pages (JSP); a new version of ASP that supports compilation of ASPs into the *Common Language Runtime (CLR)* - and therefore supports ASP scripts that use any language with an *Intermediate Language (IL)* binding.
- ❑ Win Forms and Web Forms: new UI component frameworks accessible from Microsoft Visual Studio (development environment).

Business Logic Layer:

- ❑ Component Architecture: Component Object Model Plus (COM+); the enhanced Windows Distributed Component Architecture (DCOM).
- ❑ Active Data Objects Plus (ADO+), a new generation of ADO) data access components that use XML and SOAP (Simple Object Access Protocol, see chapter 6) for data interchange.
- ❑ A set of base components, accessible from the *common language runtime*, that provide various functions (networking, containers, etc.).

# Microsoft .NET Architecture (2)

---

Business Logic layer continued...

- ❑ C#, a "new" language for writing classes and components, that integrates elements of C, C++, and Java, and adds additional features, like metadata tags, related to component development.
- ❑ *Common Language Runtime (CLR)*: Programming language-specific code is compiled into a common IL (Internal Language) bytecode format. IL code is then compiled into machine-language and run.

Storage Layer:

- ❑ Microsoft Transaction Server (MTS) for persistency of COM objects

# Comparison J2EE / .NET (1)

---

## Programming language

C# (spoken: „C sharp“) vs. Java:

- ❑ C# and Java both derive from C and C++. Most significant features (e.g., garbage collection, hierarchical namespaces) are present in both. C# borrows some of the component concepts from JavaBeans (properties/attributes, events, etc.), adds some of its own (like metadata tags), but incorporates these features into the syntax differently.

Java runs on any platform with a Java VM. C# only runs in Windows for the foreseeable future.

- ❑ C# is implicitly tied to the Common Language Runtime (CLR), and is compiled entirely into native code (just-in-time compiled bytecode is possible).

Java code runs as Java Virtual Machine (VM) bytecodes that are either interpreted in the VM, compiled to native code or Just-in-time compiled).

[Farley00]

# Comparison J2EE / .NET (2)

---

## Runtime & Distribution

IL Common Language Runtime vs. Java Virtual Machine and CORBA IDL and ORB:

- ❑ .NET common language runtime allows code in multiple languages to use a shared set of components, on Windows. Underlies nearly all of .NET framework (common components, ASP+, etc.).

Java's Virtual Machine spec allows Java bytecodes to run on any platform with a compliant JVM. In addition, CORBA allows code in multiple languages to use a shared set of objects, on any platform with an ORB available. However, CORBA is not nearly as tightly integrated into J2EE framework as COM+ is integrated into .NET.

[Farley00]

# Comparison J2EE / .NET (3)

---

## Distribution & Data Exchange

ADO+ and SOAP-based Web Services vs. JDBC, EJB, JMS and Java XML Libraries (JAXP)

- ❑ ADO+ is built on the premise of XML data interchange (between remote data objects and layers of multi-tier apps) on top of HTTP (this is called SOAP). .NET's web services in general assume SOAP messaging models.
- ❑ EJB, JDBC, etc. leave the data interchange protocol at the developer's discretion, and operate on top of either HTTP, RMI/JRMP (Java-specific remote method invocation protocol) or the Internet InterObject Protocol (IIOP).

## Server-Side Web Service Architecture

Active Server Pages+ (ASP+) vs. Java Server Pages (JSP)

- ❑ ASP+ uses Visual Basic, C#, and other languages for code snippets. All get compiled into native code through the common language runtime (as opposed to being interpreted each time, like ASPs).
- ❑ JSPs use Java code (snippets, or JavaBean references), compiled into Java bytecodes (either on-demand or batch-compiled, depending on the JSP implementation).

[Farley00]

# Comparison J2EE / .NET (4)

---

## Graphical User Interface / Web Interface

Win Forms and Web Forms vs. Java Swing:

- ❑ Win Forms and Web Forms support rapid application development (RAD) through the *MS Visual Studio* Integrated Development Environment (IDE) - no other IDE support currently available.
- ❑ Java lacks similar standardized web components (e.g., based on JSP). Some proprietary components are available through Java IDEs, etc. standardized Swing (Java-based GUI) support is available in many Java IDEs and tools.

[Farley00]

# 3. Enabling Web and Software Technologies

---

**3.1 Client / Server Architectures and the Internet**

**3.2 Internet Protocols and Internet Infrastructure**

**3.3 Multi-Tier Architectures**

**3.4 Platform Choices and Connectivity Options**

Platform Choices, Connectivity Options: ISP vs. ASP

Organizational Structures

# Platform Choices for Web Servers

---

Common web servers:

- Apache
- Internet Information Server (IIS)
- Oracle Web Server
- Jigsaw (W3C reference implementation in Java)
- Java Web Server (Sun product)

Web Servers provide only basic functionality. Extensions relevant to ECommerce are provided as **modules**.

Modules (proxy, JSP / Servlet server, distributed authoring and versioning, ...)

- must be purchased separately
- can be integrated easily into the web server

# Pricing Issues

---

Web Server	Platform	Price	Remarks
Apache	Linux, Windows NT / 2000 / 98	0 €	Open Source
Microsoft Internet Information Server (IIS)	Windows NT	N / A	Included in Windows NT Server
Oracle Web Server	Windows NT, Solaris	Examples: <ul style="list-style-type: none"><li>• 1Pr-Intel-500 Mhz: 2500 U\$</li><li>• 2Pr-SPARC-300 MHz: 4500 U\$</li></ul>	Included in Oracle Internet Application Server (iAS)

# Web Server Extension: WebDAV

---

**Web-based Distributed Authoring and Versioning** (WebDAV): Allows for multi-user, concurrent document manipulation over the Web. Some features:

- ❑ Document Editing: Editors use a check-out / check-in protocol to edit documents
- ❑ Document Folders: The WebDAV *collection* abstraction allows to map server repository directories to the Web
- ❑ Documents Versioning: supported via additional standard *DeltaV*

WebDAV supported on the client-side, some examples:

- ❑ Windows supports mapping of WebDAV collections to so-called WebFolders.
- ❑ MS Office tools have integrated WebDAV support

Technical realization: Extension of the HTTP protocol by new methods.

Integration with WebServer, e.g.,

- ❑ Integration into Apache web server via the *mod\_dav* module.
- ❑ WebDav functionality is integrated into the Tomcat Servlet Engine

For further information see [www.webdav.org](http://www.webdav.org)

# Log Files

---

Web Servers will log requests and responses in special log files (.log). They help analyze the usage of the web server / resources as well as help detect attempts on restricted areas.

Information to log:

- Request
- Response
- Date and time
- Session key (User)
- Requested resource

There is a standard log file format that can be analyzed with off-the-shelf statistics tools (commercial & Freeware).

# Web Appearance and Hosting Solutions

---

**Solution providers** are specialized service providers that connect electronic businesses to the Internet. Apart from Internet connection, they might furthermore

- Lease servers to companies
- Maintain corporate servers
- Provide standard software (e.g., Database Servers, Application Servers, Content Management Systems, Shop systems).

These services provider models are common:

- Internet Service Provider (ISP)
- Service Providers – provide Internet Services and Servers (SP)
- Application Service Provider (ASP)

Furthermore, a company can choose between different connectivity options

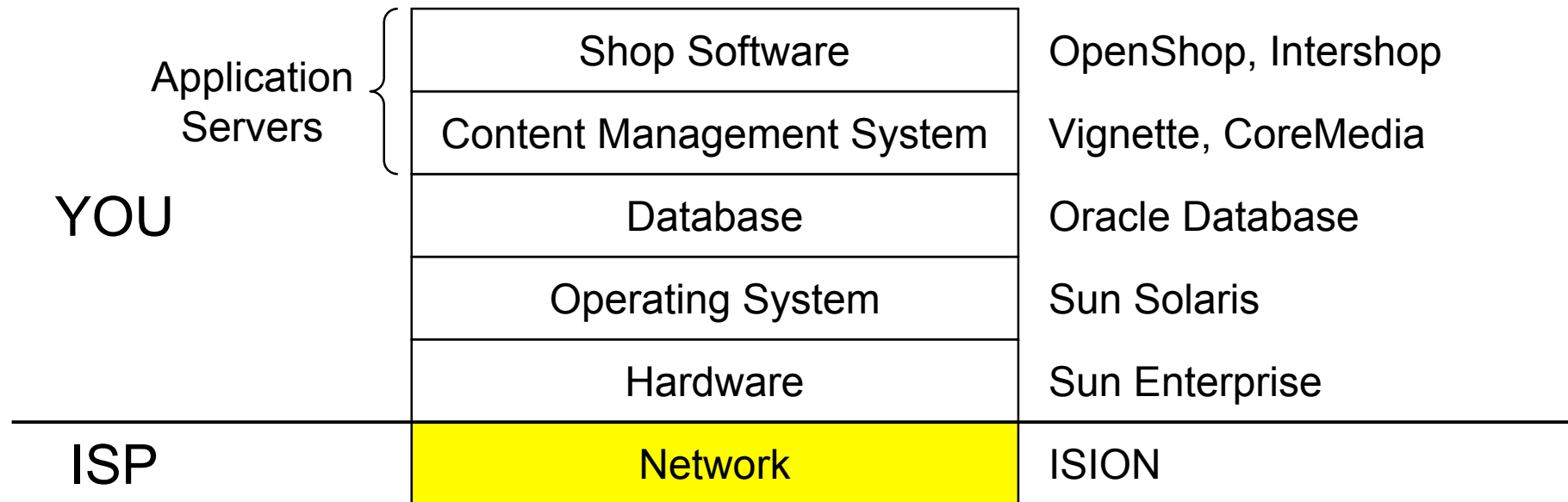
- Network services only (Internet connection only)
- Full outsourcing (Servers and Software Systems are maintained at the ASP)

# Internet Service Provider (ISP)

---

An Internet Service Provider (ISP) is a legal person / company providing network access to the Internet.

The ECommerce business company must provide everything else.



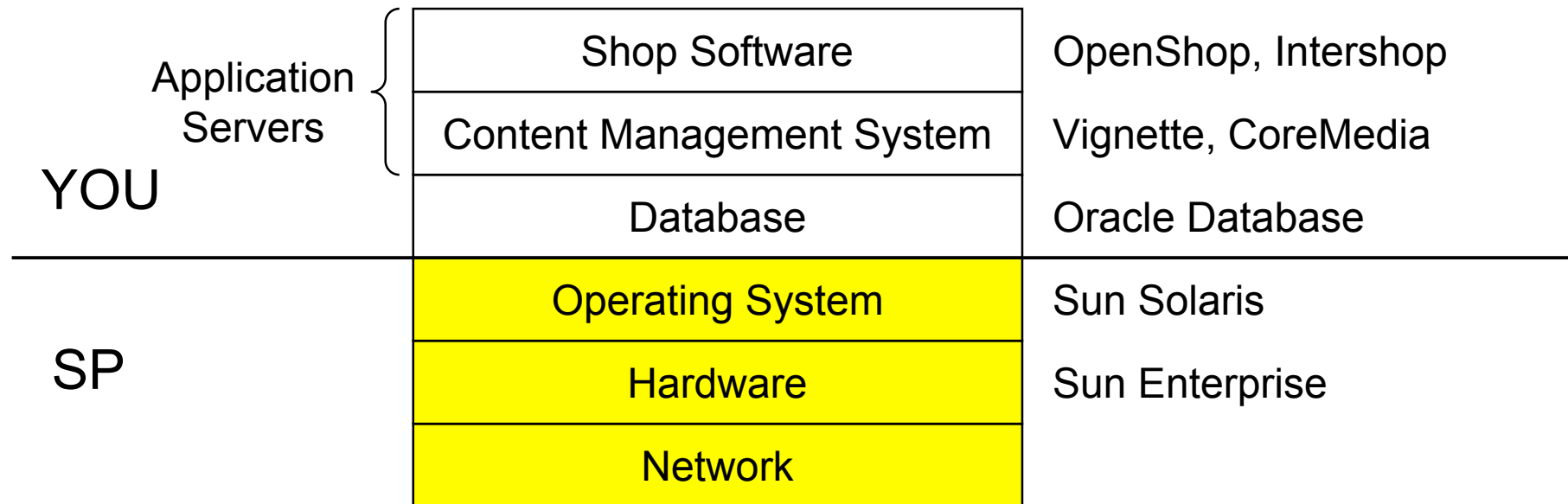
Putting up a system at an Internet Service Provider is also called *co-location*.

# Service Provider (SP)

---

The service provider provides Internet access and hardware (servers).

The ECommerce business company provides application and database software.



# Multi-Hosting (Virtual Hosts)

---

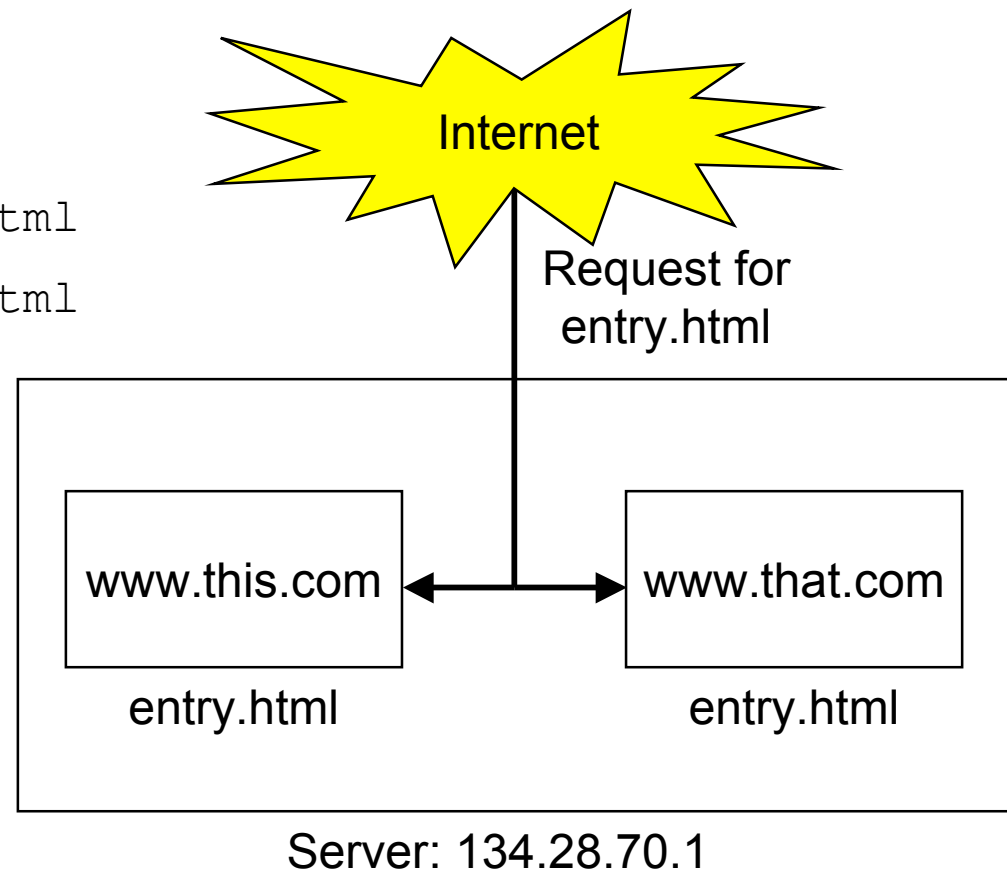
A server can manage several web hosts. Incoming requests are dispatched to the appropriate web host using the HTTP request HOST field.

This is often used by service providers to maintain several web appearances on one computer.

Example:

Host: `www.this.com:80/entry.html`

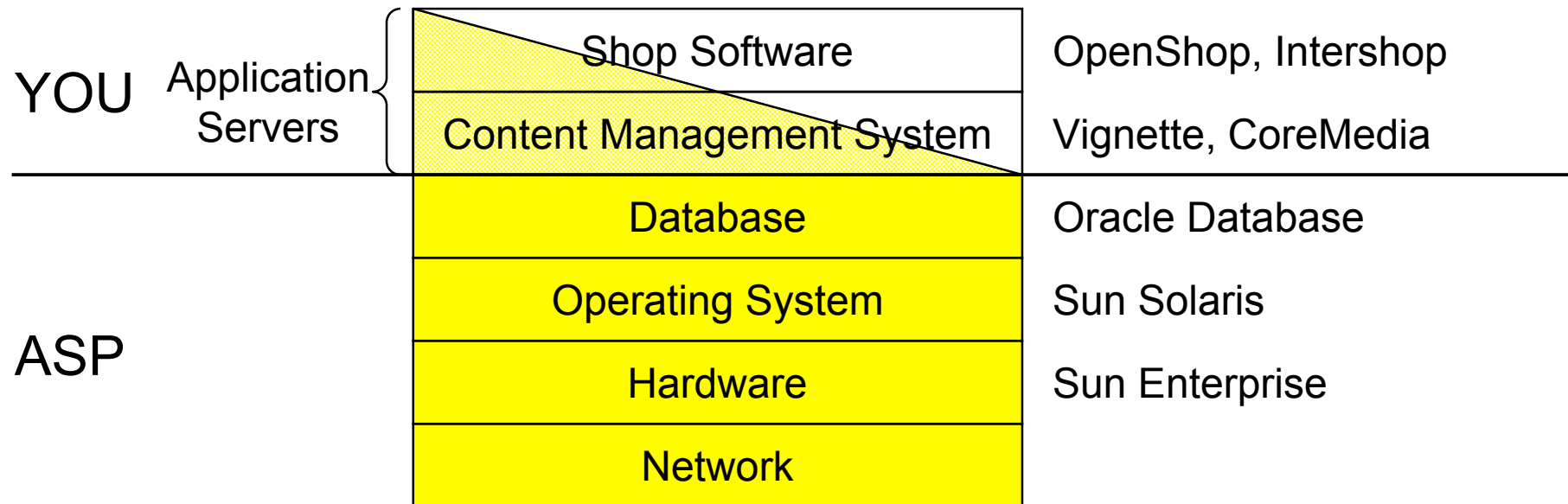
Host: `www.that.com:80/entry.html`



# Application Service Provider (ASP)

---

The ECommerce business company may provide additional application executables.  
Application Service Provider (ASP) provides everything else.



Examples for SP and ASP solution providers:

❑ [www.hostingsolutions.com](http://www.hostingsolutions.com), [www.netfirms.com/](http://www.netfirms.com/)

# ASP Shop Solutions

---

Low-cost online shop / mall solutions usually comprise:

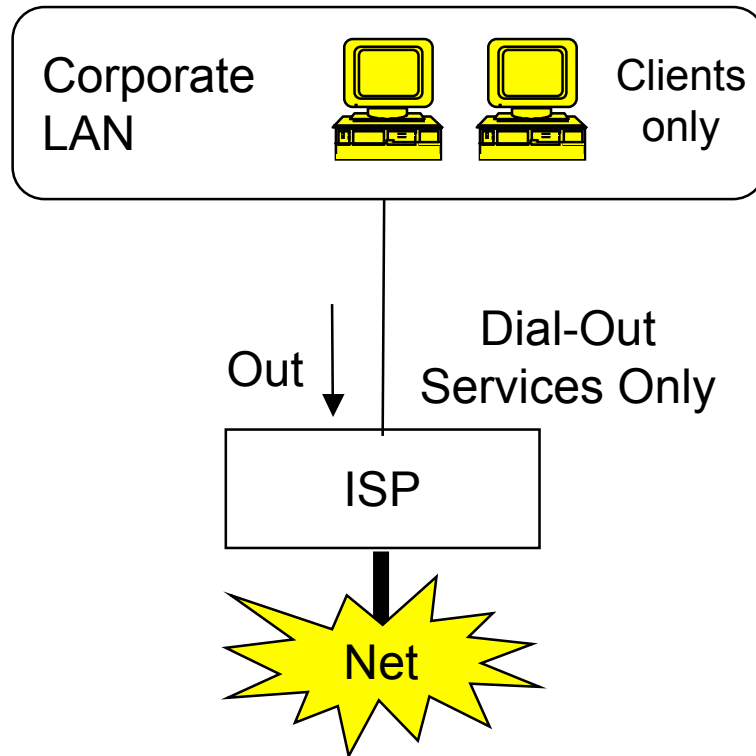
- Domain name: Usually setup with an email address
- Shop homepage + authoring tools: ASP provides authoring tools for web content creation. Offline web content can be deployed to ASP server.
- Inclusion of multimedia content: Graphics, audio and video clips, server-side includes.
- Statistics: Functionality to retrieve statistical information about shop.
- Basic online product catalogs: Usually hierarchical product categories. Exclude product variants.
- Shopping cart functionality
- Order functionality: Merchant is notified of orders by mail. Order processing done by merchant.

Usually such solutions do not comprise

- Credit-card validation and payment processing
- Integration of payment processing with financial software
- Transaction tracking

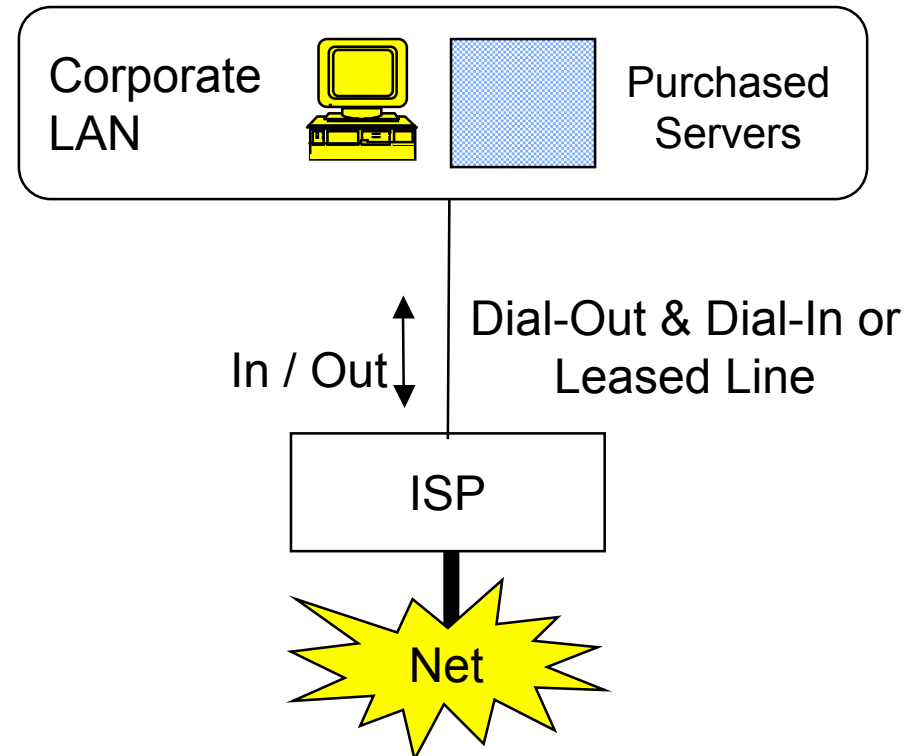
# Business Connectivity Options: ISP Solutions

1a) Network Services for corporate clients only



**Lowest Cost**

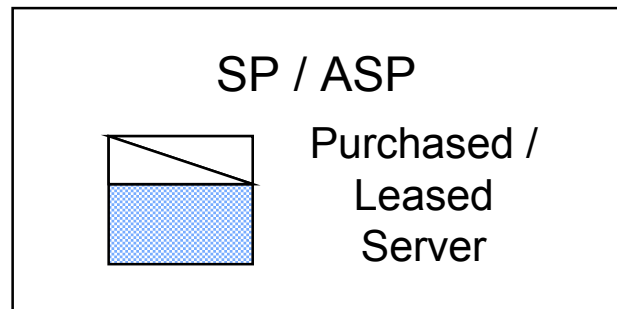
1b) Network Services only



**Easy maintenance,  
suitable for low  
volume services**

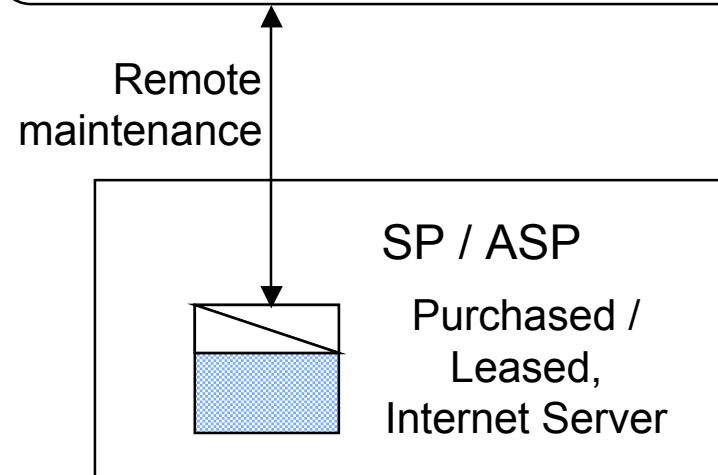
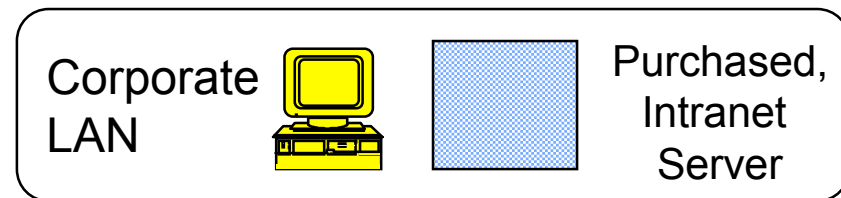
# Business Connectivity Options: SP / ASP Solutions

## 2) Full outsourcing



**No up-front investment,  
low risk**

## 3) Mixed solution



**Keep servers  
close to their users**

# Internet, Intranet and Extranet (1)

---



**Definition**

**Intranet:** Computers inside a company that are connected via internet-based protocols (HTML: HTTP, FTP, Mail: SMTP, POP3) and are not accessible from outside.

Why an intranet is not sufficient:

- ❑ A company that has several local branches or has field staff needs to provide access to its intranet to externals. Furthermore, a company may want to provide access to internal information to its business associates.

Example: Boeing subcontractors must allow Boeing access to their workflow systems for inter-organizational workflows.

Solution: Creating an **extranet**.

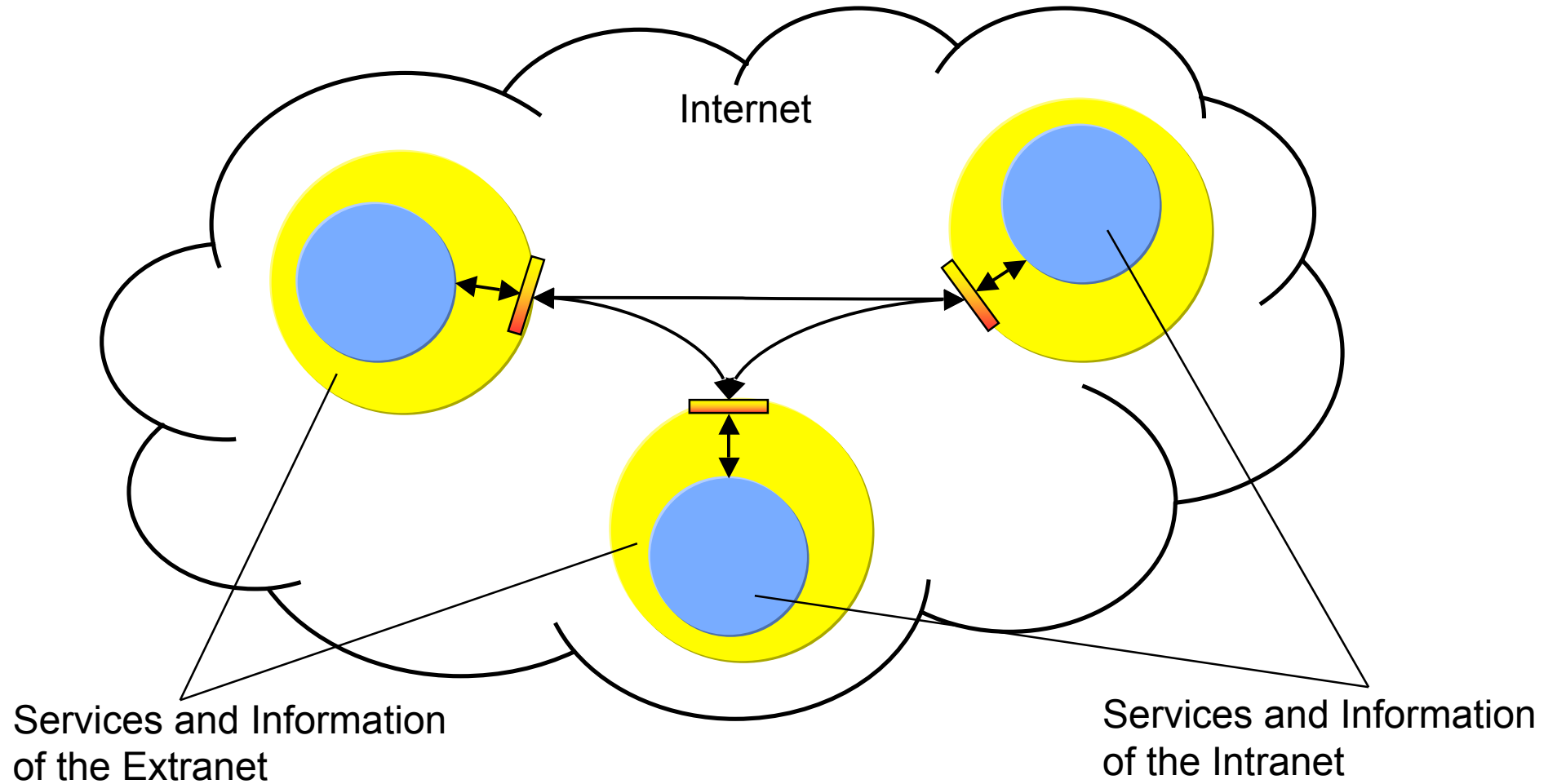


**Definition**

A company's **extranet** provides secured access to its intranet and to additional information and services that may not be part of its intranet. Access to the extranet is secured via a firewall.

# Internet, Intranet and Extranet (2)

---



# Internet, Intranet and Extranet (3)

Classification [Merz99]

<b>Access to the company's network</b>	<b>Internet</b>	<b>Intranet</b>	<b>Extranet</b>
Predominant use for:	public, customers	employees	employees, business associates, subcontractors
Security	No authentication, no authorization	Authentication, Authorization	Authentication, authorization, non-repudiation (traceability)
Online-Community services	Mailing lists, Chat, Search Engines,	Mailing lists, groupware, workflow systems, knowledge management systems	Mailing lists, inter-organizational workflows. knowledge management
Data exchange	Marketing information, Product information	Employment lists, document repositories, administrative data	tender catalogs, pricing lists, terms and conditions, product information, distribution information, XML / EDI

# Virtual Private Networks (VPN) (1)

---



**Definition**

A **Virtual Private Network** is a network that is constructed by using public wires (the Internet) to connect network nodes, such as distributed enterprise departments, etc. VPNs use encryption and other security mechanisms at their borders to ensure that only authorized users can access the network and that the data cannot be intercepted.

[Web00]

VPNs are used to

- connect a company's extranet with its field staff over the internet. Data transfer is encrypted, access requires authentication.
- connect several companies' extranets over the Internet.

Requirements

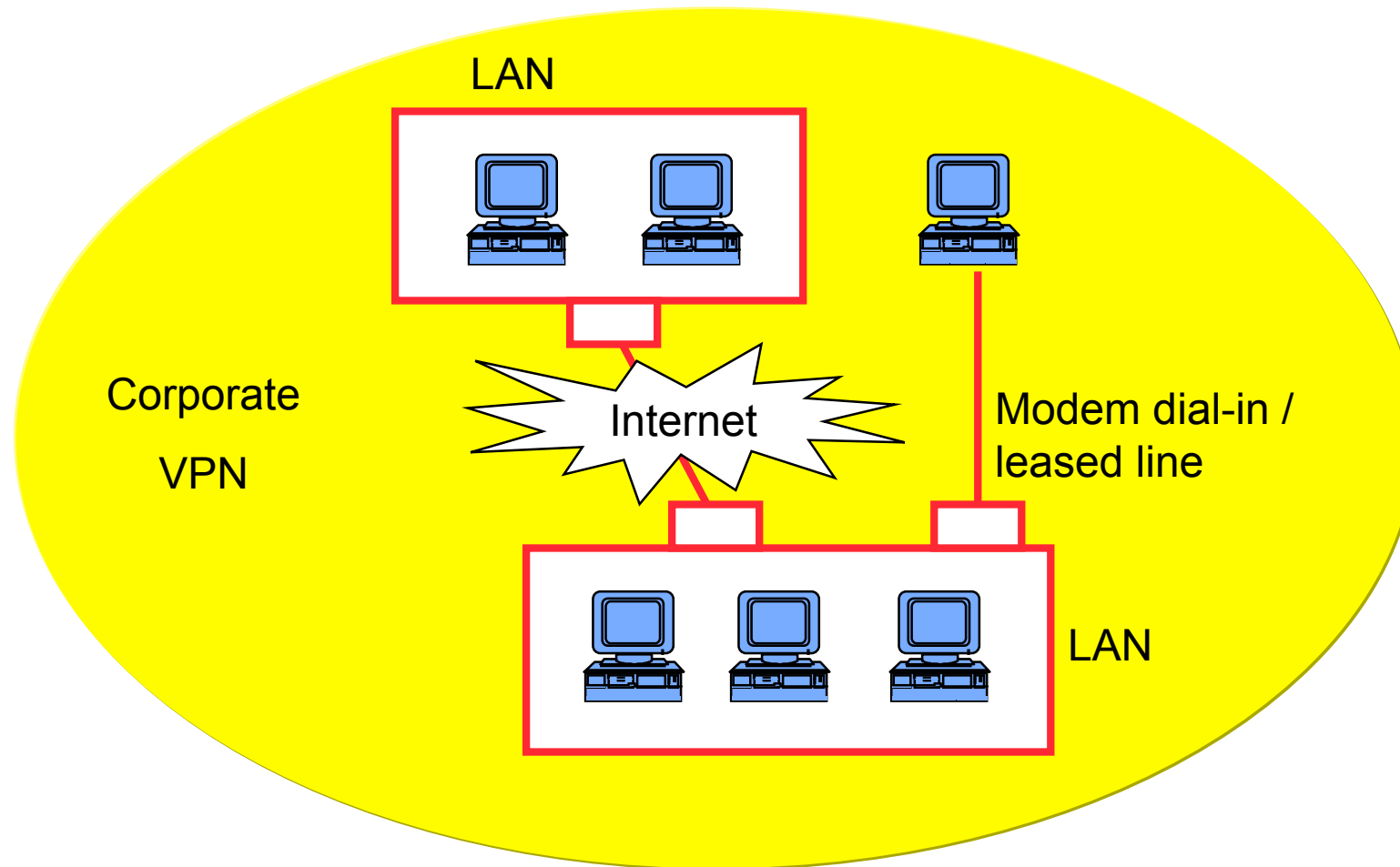
- Security measures for the extranet
- Encrypted data exchange

NOTE: In a VPN with several parties involved, every two companies will want to create a VPN on their own. In this case, a public key infrastructure (PKI) becomes useful.

For Public Key Infrastructure (PKI) see later chapters

# A Single Corporate Virtual Private Network

---



# Virtual Private Networks (VPN) (2)

---

## Realizations of a VPN:

by Black boxes: Devices that are plugged between computer systems and LAN connection.

- + Independent of hardware systems and operation system
- + Security services for systems that cannot integrate security internally (e.g., “dumb” terminals)
- + Application independent
- + Ease of maintenance.

by Security (sub)layer in end systems. This is hardware and operation system-dependent. The security (sub)layer is integrated transparently into the Internet protocol layer.

- + Cheaper solution
- + End-to-end security

NOTE: As VPNs require the management of numerous keys/passwords, there is an interest to set up a *public key infrastructure* (PKI), see chapter 5.