

INB/INN 373

Web Application Development

What's this Unit About?

- Software architecture:
 - N-tier - architecture for web based systems.
- Technology:
 - Learn about technologies used to create web based application
 - Some ancient, some bleeding edge.
 - Some open, some proprietary.
 - Some cross platform, some platform specific.
 - Learn to critically analyse technologies
 - Compare and contrast
 - Plan for and predict the future
- Overcoming real-world challenges
 - Diagnosis, Self directed research, Perseverance

Lecture Schedule

Week	Topic	Lecturer
1	Introduction to web based systems	JZ
2	Client-side processing	JZ
3	Server-side processing	JZ
4	Introduction to the .NET platform	JZ
5	ASP.NET and the Presentation Tier	JZ
6	The Middle Tier - pulling it all together	JZ
7	ADO.NET and the Data Tier	JZ
8	Web Application Security	JZ
9	Web Services	JZ
10	AJAX	JZ
11	Other Development Options – J2EE, Rails, LAMP	TBA
12	Web Content Management Systems	JZ
13	New Developments and Review	TBA+JZ

Practical Schedule

Week	Topic
Week1	Self Exercise - Install Free Software on your own computer <ul style="list-style-type: none">• Visual Studio 2012 Ultimate, MS .NET Framework 4.5• SQL Server 2012 Developer• Windows 7, Professional Edition, IIS 8 Express
Week2	Introduction to Visual Studio and C#/VB.NET
Week3	Introduction to HTML Forms and JavaScript
Week4	Introduction to Server Side Processing (IIS, ASP etc)
Week5	Introduction to .NET
Week6	Introduction to ASP.NET
Week7	Introduction to Data Source Objects and the Business Tier
Week8	Introduction to ADO.NET
Week9	ASP.NET Forms Security
Week10	INB373 Assignment Upload Tutorial

Teaching Staff

■ Unit Coordinator, Lecturer

- Dr Jinglan Zhang
- 3138 9353
- Jinglan.Zhang@qut.edu.au
- Office: S Block Level 10 1037
- Consultation: by appointment, please phone or email to arrange a suitable time, preferably 9-10am Tuesdays and Thursdays

■ Tutor

- Mr Anthony Truskinger
- 3138 9340
- anthony.truskinger@gmail.com
- Office: S Block Level 10 1006
- Consultation: by appointment, please phone or email to arrange a suitable time

Administration

Prerequisite:

- INB271/INN271 or equivalent
simple web development, simple programming, simple database design
- INB/INN270 or equivalent

Contact Hours per week:

- **2 hour Lecture**
- **2 hour lab session (commencing in week 2)**

Textbooks:

- *None*

References:

- See lecture notes for lists of online references.

Assessment:

- | | |
|--|----------|
| ■ Homework exercises (pass/fail, marked in tutorials) | 10 x 1 % |
| ■ ASP.NET Programming Assignment (groups of 1, 2 or 3) | 40 % |
| ■ Final Exam | 50 % |

How to Survive this unit?

- Attend lectures
- Complete all practicals
- Work with an assignment partner
- Sit in final exam

Study Hours

- A 12 credit point unit implies that an average student will need to spend an average of 12 hours per week to pass this unit.
- Weaker students or students wishing to do better than simply pass may need to spend more hours per week.

- As a rough guide, I recommend that you allocate your time approximately as follows:
 - Attending Lectures:
 - Attending practicals:
 - Weekly pre-reading/reviewing lecture notes
 - Textbook and other readings
 - Assignment
 - Start revising for final exam

$13 \times 2 \text{ hours} = 26 \text{ hours}$
 $12 \times 2 \text{ hour} = 24 \text{ hours}$
 $13 \times 1.5 \text{ hours} = 19.5 \text{ hours}$
 $13 \times 1.5 \text{ hours} = 19.5 \text{ hours}$
 $8 \times 7 \text{ hours} = 56 \text{ hours}$
 $1 \times 11 \text{ hours} = 11 \text{ hours}$

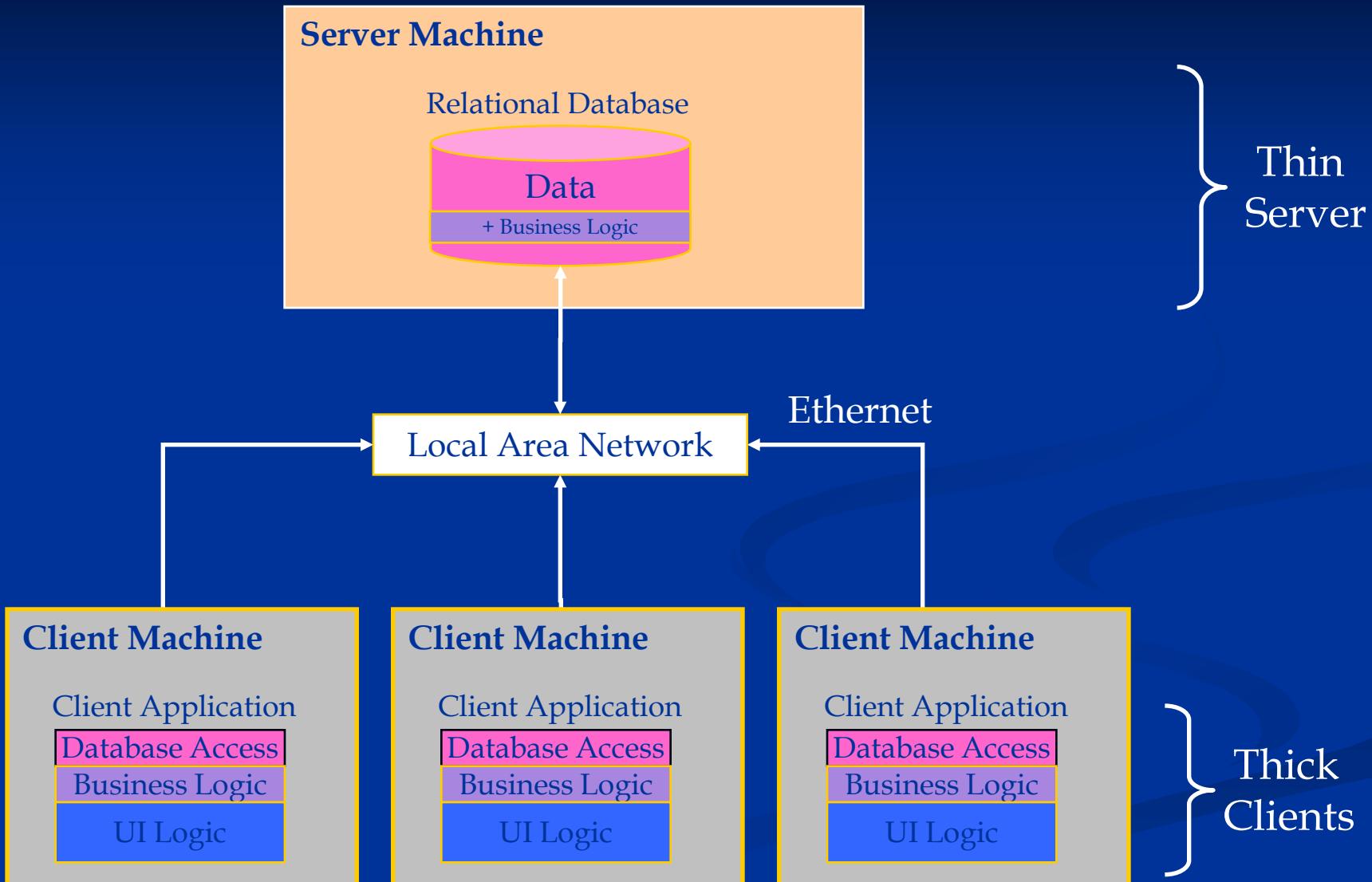
 $13 \times 12 \text{ hours} = 156 \text{ hours}$

- If you find these tasks taking considerably more time than this, please notify the unit coordinator.

Failing Students from 2010

	Homework (worth 10%)	Assignment (worth 40%)	Exam (worth 50%)	Total
Mike	30%	43%	57%	49%
Amber	30%	43%	57%	49%
Ralph	40%	65%	33%	47%
Joyce	40%	55%	40%	46%
Edward	10%	43%	52%	44%
Victor	10%	75%		31%
Samuel	20%	70%		30%
Harry	50%		40%	25%

Client-Server Applications



Client-Server Applications

■ Advantages:

- Access to centralized data.
- Rich GUI on the client machines
- Client-side validation and processing possible.
- Some central enforcement of business rules

■ Disadvantages:

- Difficult to deploy and upgrade client applications.
- Distributed enforcement of business rules.
- Requires homogeneous client platform.
- Business rules implemented in SQL.
- Limited to local area networks.

Then along came ...

The Internet:
A Network of Networks
(1969)

A Brief History of the Internet

1957 Russians launch Sputnik

- Eisenhower creates Advanced Research Projects Agency (ARPA)

1960 First commercial computers

- Point to point communication using circuit-switched telephone lines.

1962 Packet-switching networks proposed

- MIT PhD student Leonard Kleinrock

1969 ARPA creates first WAN using Packet switching (ARPANet)

- 4 Nodes (UCLA, Stanford, UC Santa Barbara, Univ of Utah)

1972 First public demonstration of ARPANet

- 40 Nodes, supports email and telnet

1973 International nodes join ARPANet (England and Norway)

- Ethernet for LANs, Gateways and FTP developed

1975 ACSNet created in Australia (ftp, email & news)

- dialup modem-based network with store-and-forward

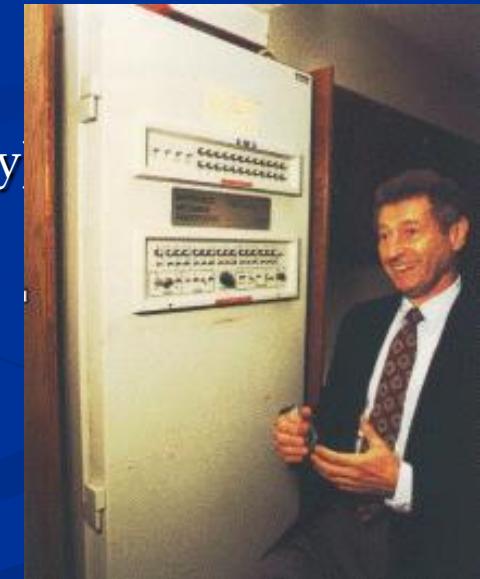
1982 ARPANet moves from NCP to TCP/IP protocol suite

- The term “Internet” introduced

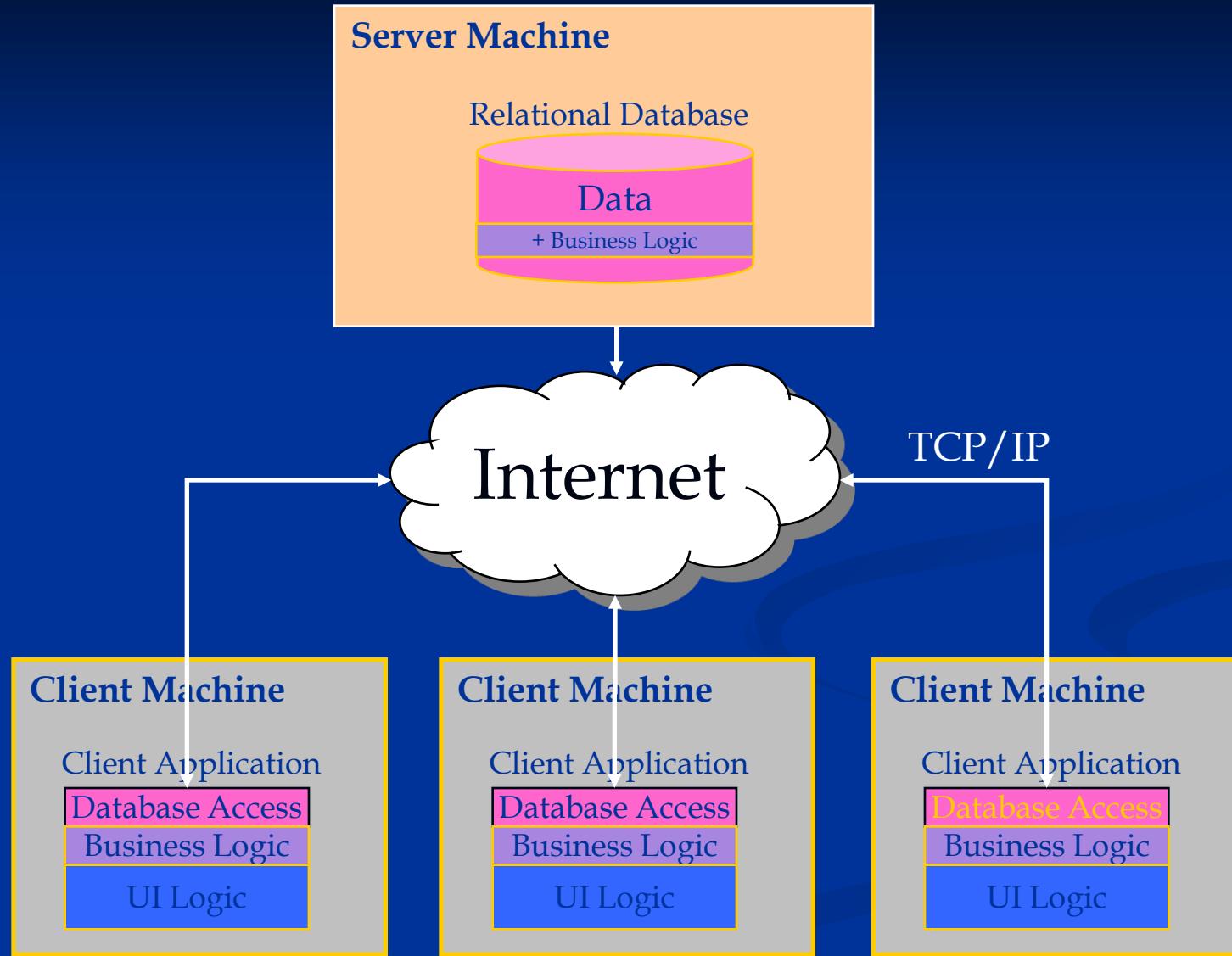
1984 Domain Name Servers (DNS) developed

- 1000 Nodes, 131.181.111.1 becomes www.scitech.qut.edu.au

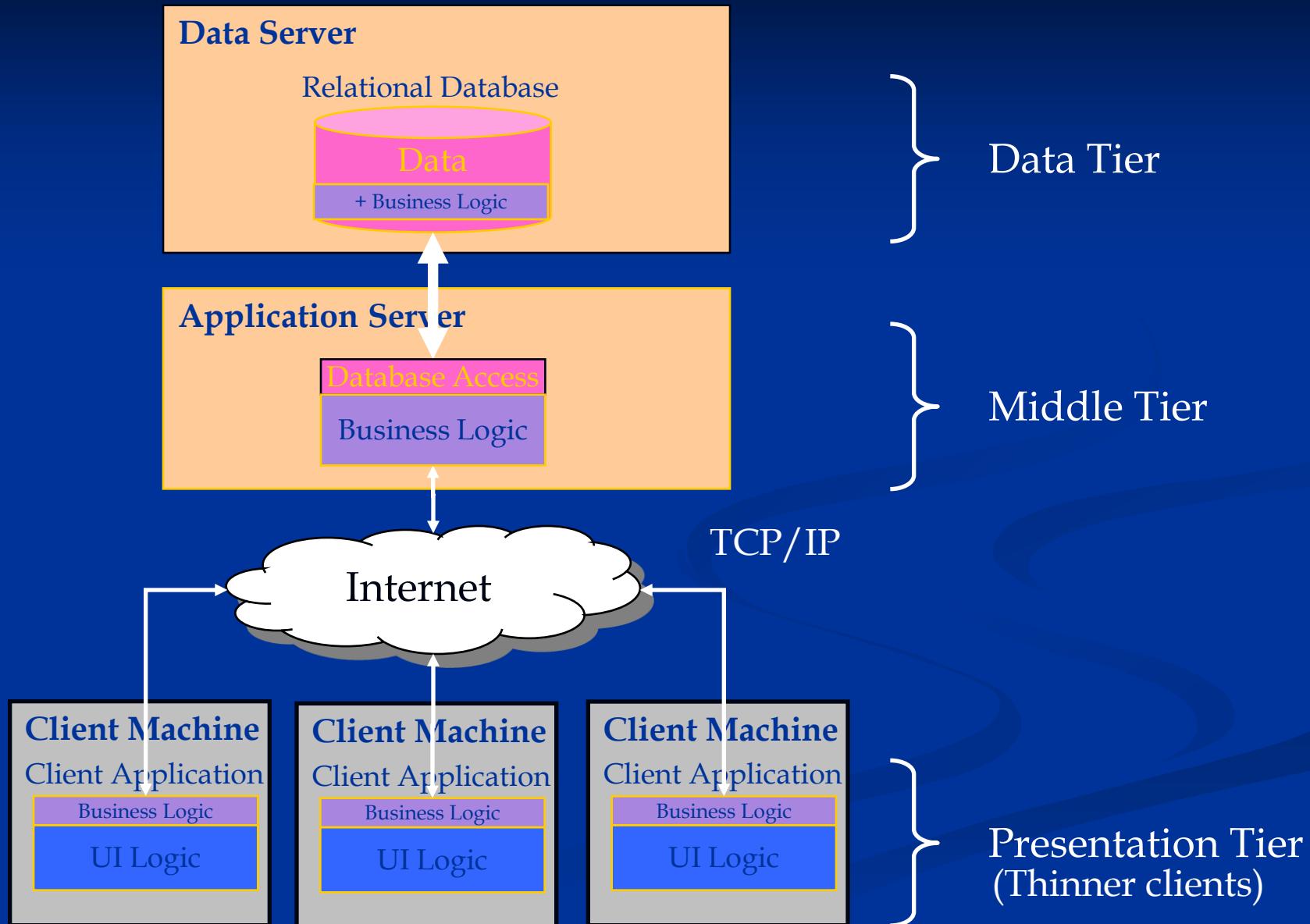
Later trend: speed (56k modem to ADSL), wireless, handheld devices



Internet Client-Server Applications



3-Tier/N-Tier Architecture



3-Tier/N-Tier Architecture

■ Advantages:

- **Flexible and reusable application**
 - Separates UI from business logic.
 - Central enforcement of business logic.
 - Easier to update business logic.
- **Different tiers can run on physically separate platforms.**

■ Disadvantages:

- Difficult for TCP traffic to penetrate Firewalls.
- Still difficult to deploy and update client applications.

Then along came ...

The World Wide Web

(1990)

A Brief History of the Web

1960 Harvard PhD student Ted Nelson invents computer-based Hypertext

1985 Standard Generalized Markup Language (SGML)

1990 World-Wide Web (WWW) project

- Created by European Organization for Nuclear Research (CERN)
- HTML, URL's and HTTP



1993 NCSA releases Mosaic (first graphical web browser)

Common Gateway Interface (CGI) - first support for web-based applications

1994 Mosaic commercialized by Netscape

1995 HTML Forms standardized

Apache web server 1.0 - based on popular NCSA web server

Sun announces "Java to be incorporated into Netscape"

1996 Microsoft announces that it will "embrace and extend" the Internet

- Internet Information Server 1.0
- Internet Explorer 1.0
- Active Server Pages (ASP) 1.0

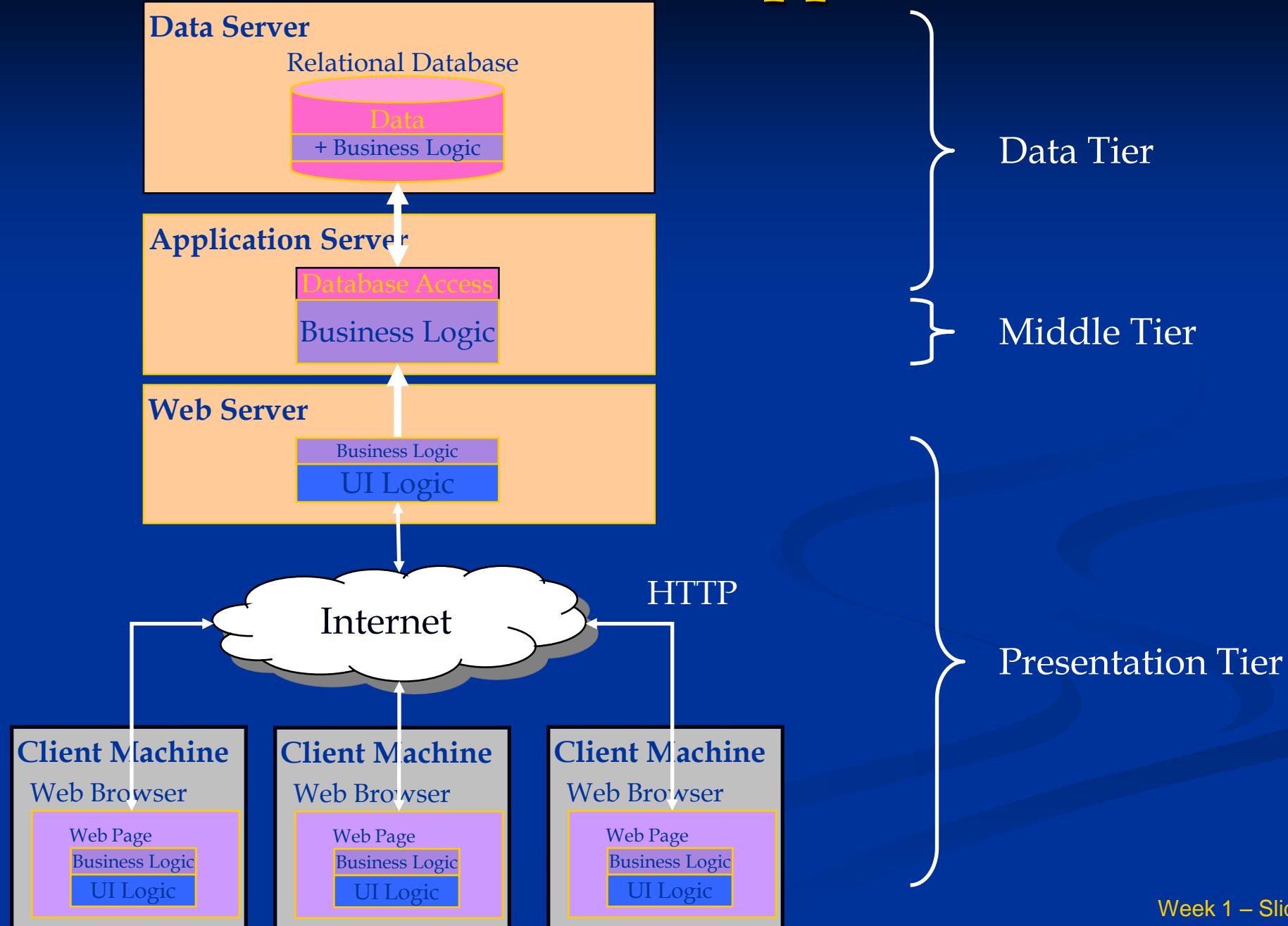
1997 Sun's Java Servlets

1999 Sun's Java Server Pages (JSP)

2000 Sun's Java 2 Platform Enterprise Edition (J2EE)

2001 Microsoft's ASP.NET

Web-based Applications



Web-based Applications

■ Advantages:

- Much easier to deploy and upgrade client software.
- Allows heterogeneous client platforms

■ Disadvantages:

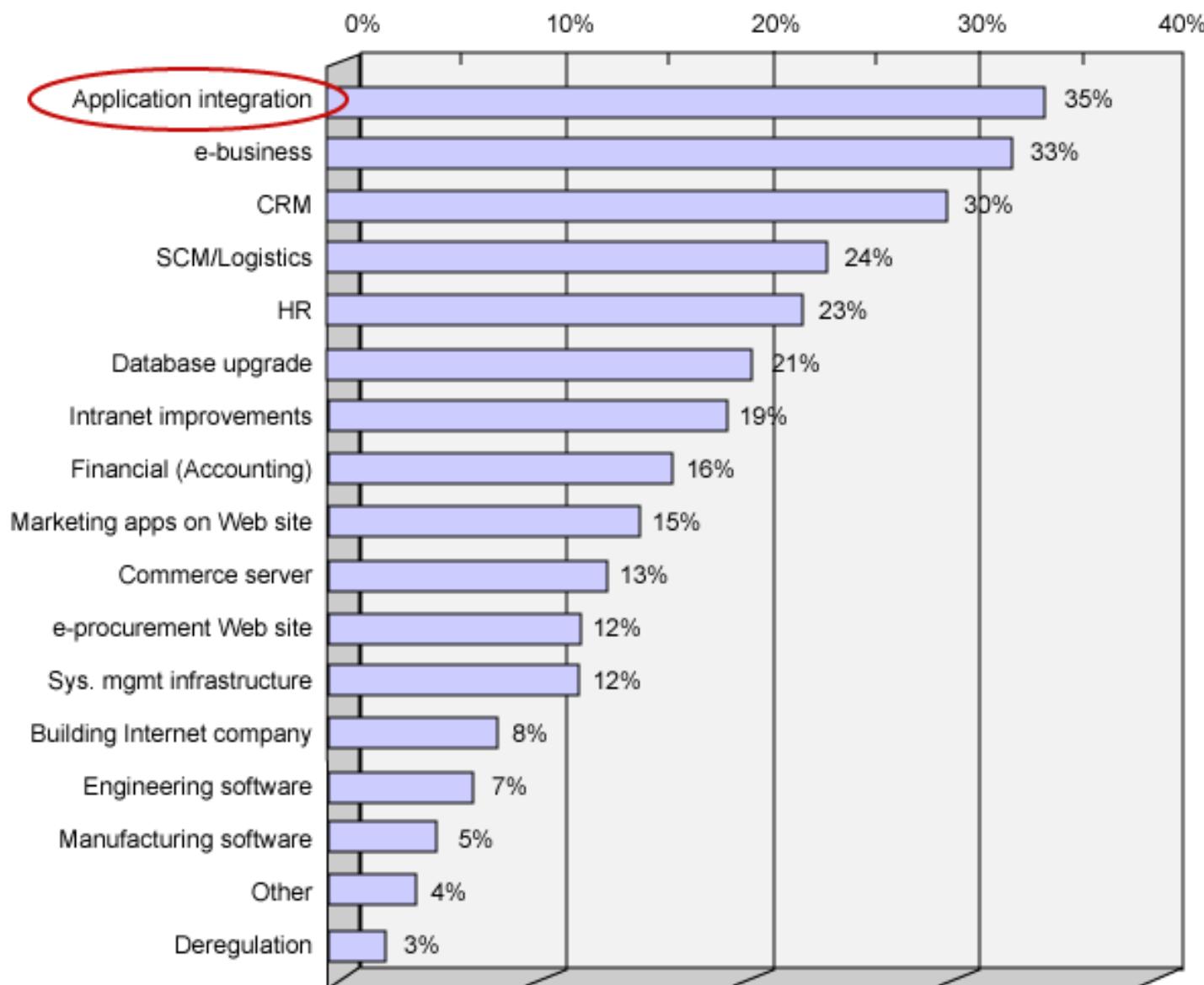
- Browser hosted GUI is not as rich as a thick client (limited by HTML).
- Any client-side processing is interpreted script-code (not as expressive and efficient as compiled code).

Beyond N-tier Systems: Application Integration

- Large organizations need to maintain and use many different software systems.
 - Eg: Invoice processing, Inventory control, Payroll and Online sales.
- Some of these might be web-based systems, others not.
- Ideally, these systems need to be *integrated*, i.e. they at least need to be able to *communicate* with one another.
- This is arguably the biggest challenge facing the IT industry.
 - Failure to integrate an organization's computer systems can severely hamper its ability to do business.
 - But such integration is not easy, and is currently consuming much of the resources of many organization's IT sections.

Top strategic software platform project over the next year

% of respondents



Application Integration using Web Technologies

- Web technologies are being used not just to deliver application functionality to end users, but also to integrate business applications with one another.
- Key idea is to loosely couple systems using platform and language independent open standards such as XML and HTTP.

Web Services

- Web servers directly expose functionality rather than a user interface.
 - allows the facilities provided at a remote web site to be accessed programmatically.
- Similar to remote procedure calls, but encoded using XML/JASON and transported using HTTP
 - language and platform independent.
 - Broad vendor support for web service standards.
- Easy to create distributed applications from components created and deployed remotely by other organisations.
 - the basis for business to business (B2B) systems.

Unit Summary

	Client-side Content (Week 2)	Web Browser	Web Server	Server-side UI (Week 3)	Business Logic (Week 3)	Integration	Data Access	Databases (Week 7)
Microsoft .NET (Week 4)	VB Script JScript	IE	IIS	ISAPI ASP ASP.NET (Week 5)	COM+ .NET (Week 6)	MSMQ Remoting	ODBC ADO ADO.NET (Week 7)	Access SQL Server
Sun Java Netscape	JavaScript	Netscape		Servlets JSP	EJB J2EE	Java RMI	JDBC	
W3C Other	HTML ECMAScript	Opera	Apache	CGI		SOAP Web Services (Week 9)		Oracle DB2

Developing Web-based Applications -Software Engineering

- Like all major software engineering projects:
 - Project planning, management and quality assurance
 - Need to work effectively in teams
 - Iterative and Incremental software lifecycle
 - Analysis, Design, Implementation, Testing and Deployment
 - High-quality coding practices:
 - Thorough documentation?
 - Object and component oriented design and programming
 - Respecting encapsulation and abstraction
 - Clear and elegant code
 - Separation of user interface from application logic

Why is Software Development for the Web more difficult?

- New technologies to learn and decide between:
 - CGI, ASP, JSP, HTML forms, JavaScript, XML, .NET, J2EE, WSDL, REST
- Client-Server architecture:
 - When and where does processing occur?
 - How do the pieces talk and work together?
- Heterogeneous client platforms and browsers
 - Internet Explorer, Mozilla, Mobile phone
- Network communication
 - Unreliable, Insecure, High latency
- Deployment and Performance
 - Can't be sure exactly how many concurrent users to expect.
 - Performance should degrade gracefully as load increases.
 - Need to be able to scale performance by adding additional servers.
 - Needs to be available 24 hours a day, 7 days a week.
 - Elegant fail over to backup servers

Microsoft's.NET platform

- .NET is a virtual execution environment.
- Similar in spirit to the Java Virtual Machine, but:
 - not platform independent and
 - not language specific.
- New Microsoft languages C# and Visual Basic.NET
 - but also supports third party languages
(eg COBOL.NET, Gardens Point Component Pascal, Ruby.NET)
- Supported by Visual Studio IDE
 - Usable by all languages
- New component model
 - Replaces COM
- Common class library
 - Accessible from all languages
 - including a new library for creating GUI applications
 - ASP.NET is one such class library ...

C# and VB.NET Readings

MSDN Library (<http://msdn.microsoft.com/library>)

- Developer Tools and Languages
 - Visual Studio .NET
 - Visual Basic and Visual C#
 - Reference
 - Visual C# Language
 - C# Language Tour
 - C# Language Specification
 - C# Programmer's Reference
 - C# Tutorials
 - Visual Basic Language
 - ...