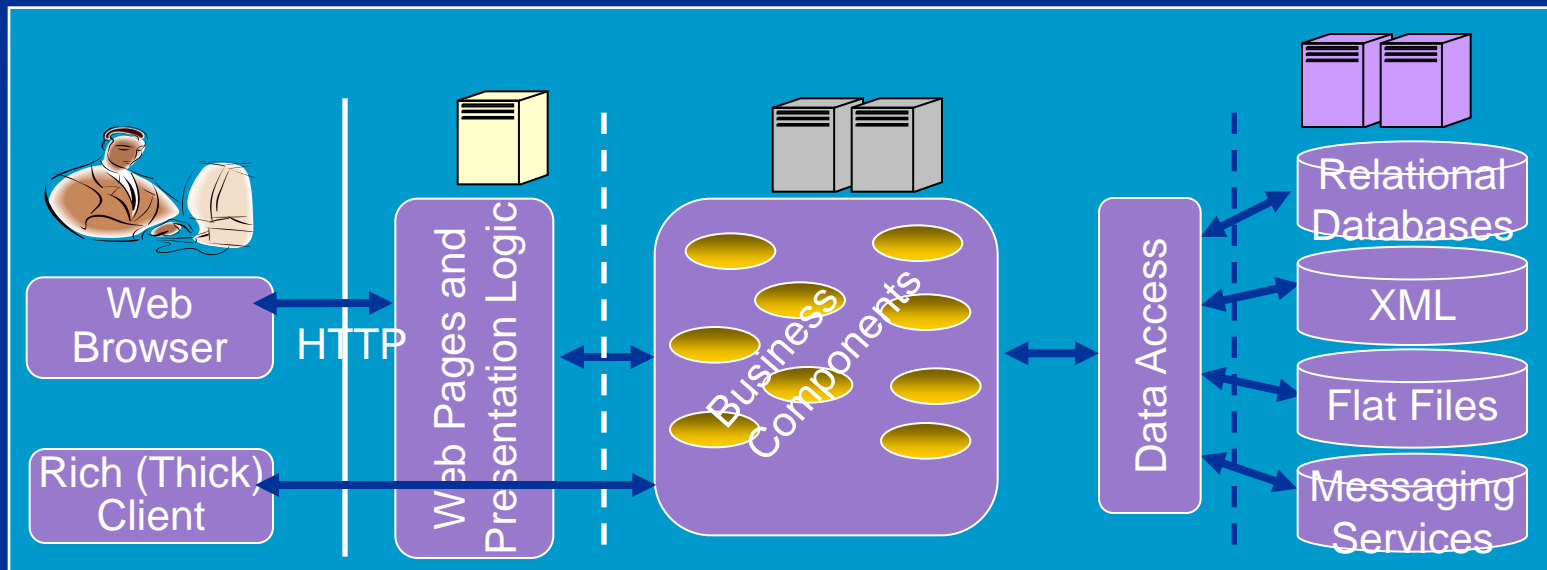# Web Application Development

Where to from here?

Beyond N-tier Systems

A Web of Services

Combination of specific services provided business partners and generic publicly available web services

Human client with web browser

Web Page Interface

HTML HTTP

Web Application

SOAP HTTP

Web Application

SOAP HTTP

SOAP HTTP

Web Application

Web Service Interfaces

SOAP HTTP

Web Application

SOAP HTTP

Human client with rich desktop app.

SOAP HTTP

SOAP HTTP

Web Application

HTML HTTP

Human client with web browser
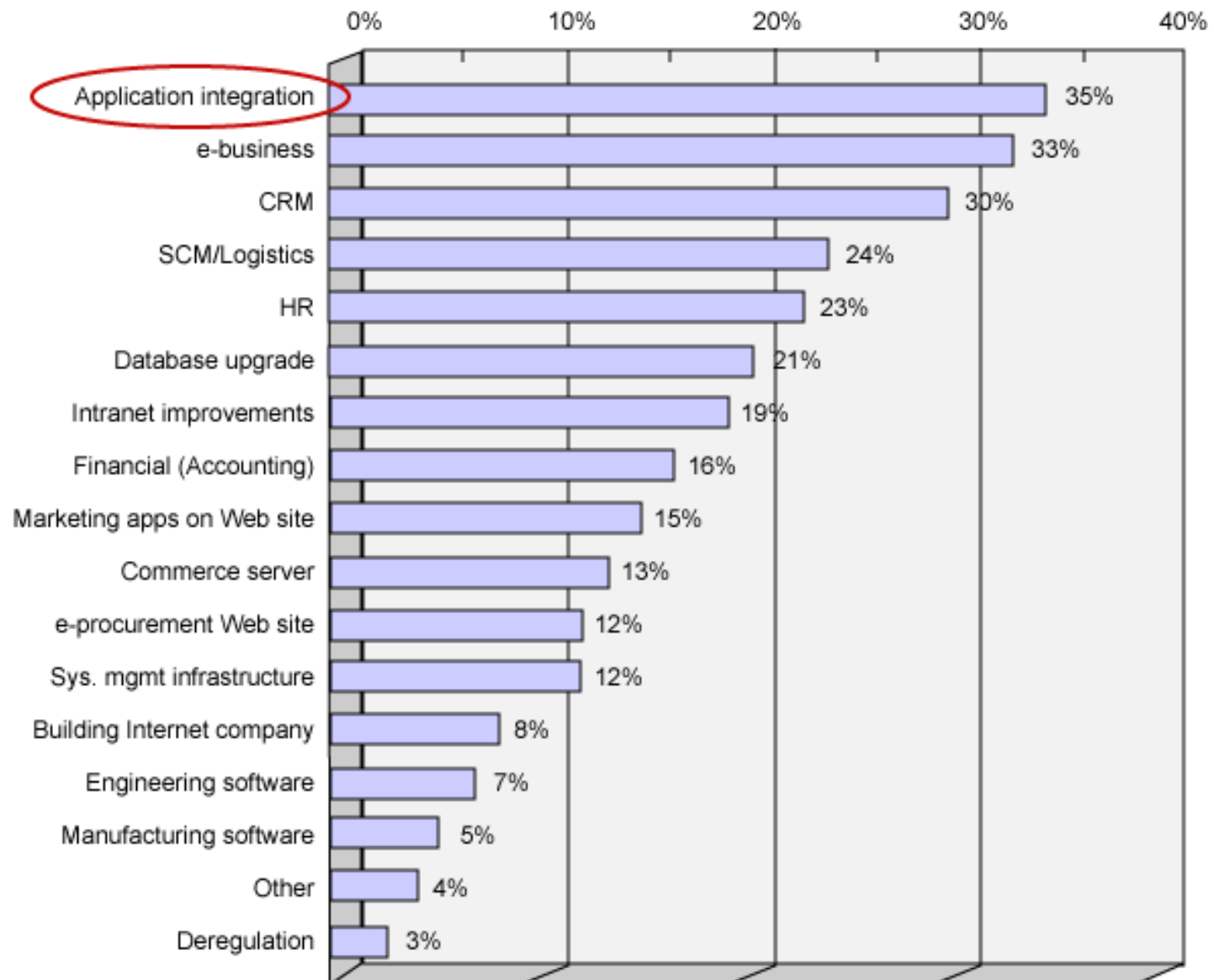
Service Oriented Architecture

- A service is a function that is well-defined, self-contained, and does not depend on the context or state of other services .

- A service-oriented architecture is a collection of services that communicate with one another.

- Web services support a service-oriented architecture, but a service oriented architecture doesn't necessarily need to use web services.

Integration

- Enterprise Application Integration (EAI)

- Business to Business Integration (B2B)

- Business Process Automation

Top strategic software platform project over the next year
% of respondents

| Category	%
Application integration	35%
e-business	33%
CRM	30%
SCM/Logistics	24%
HR	23%
Database upgrade	21%
Intranet improvements	19%
Financial (Accounting)	16%
Marketing apps on Web site	15%
Commerce server	13%
e-procurement Web site	12%
Sys. mgmt infrastructure	12%
Building Internet company	8%
Engineering software	7%
Manufacturing software	5%
Other	4%
Deregulation	3%

Enterprise Application Integration (EAI)

- Organizations need to integrate different software systems
  - Eg: Invoice processing, Inventory control, Online sales.
- Essential, but difficult and expensive to achieve.
  - developed and evolve independently
  - different source languages and platforms
- The Solution:
  - Service-oriented architecture.
  - Loose coupling (minimize complexity of interfaces)
  - Use open standards such as XML
- For more information see INB374 Enterprise Software Architecture

The Web 2.0 Manifesto

1. **The Web As Platform**
   - E.g. Google
2. **Harnessing Collective Intelligence**
   - E.g. Google page rank, Wikipedia, Flickr, Blogs
3. **Data is the Next Intel Inside**
   - E.g. Google indexes, Facebook, Google maps
4. **End of the Software Release Cycle**
   - Continual upgrade of live services
5. **Lightweight Programming Models**
   - E.g. REST vs. SOAP
6. **Software Above the Level of a Single Device**
   - Not tied to PC, e.g. iTunes
7. **Rich User Experiences**
   - AJAX, PC like interactivity, e.g. Gmail

Cloud Computing
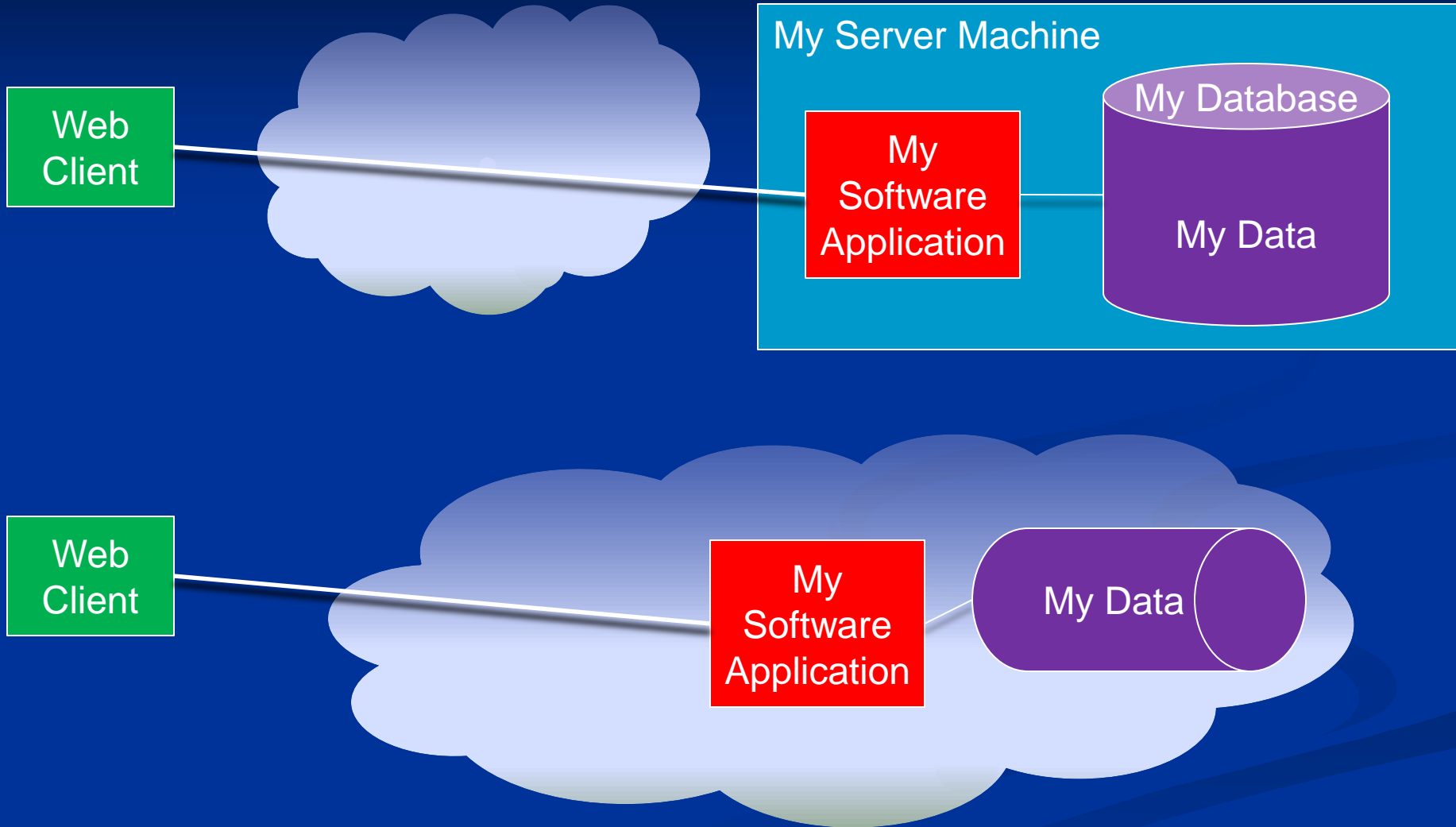
Cloud Computing Principles

- Software as a Service + Utility Computing = Scalable hosted Services
- Rational: Servers are expensive to maintain
  - Rapidly responding to changing demands is very difficult.
  - Economies of scale through mass hosting
- Key Technology: Virtualization => Elastic resources
- Requirement: Close to cheap energy, cooling and fat data pipes
- Business Model:
  - Pay as you go, Pay only for what you use
  - Very low entry costs, encourages small entrepreneurs
- Even traditional client-side apps can be hosted in the cloud
  - e.g. GoogleDocs.
- Enables thinner clients
  - mobile phones, etc.

Major Cloud Providers

- Amazon Web Services
  - Elastic Compute Cloud (EC2)
- Microsoft
  - Azure
- Google
  - AppEngine and BigTable
  - Google Docs
- Salesforce.com

Amazon Web Services

- Elastic Compute Cloud (EC2)
  - Provides you with a set of computers and file storage.
  - You choose: how many computers, operating system, application software, database, etc.
  - General purpose, but very low level.
- Amazon SimpleDB
  - Non-relational database – each item has a set of attribute values.
- Amazon Simple Storage Service (Amazon S3)
  - File-like data storage ("data objects" are stored in buckets)
- Amazon CloudFront
  - Provides fast cached access to content in Amazon S3
- Amazon Simple Queue Service (Amazon SQS)
- Amazon Elastic MapReduce

The Microsoft Cloud Platform

- Azure
  - Cloud Computing Operating System
  - Commercial offering in 41 countries (incl Australia)
  - Two types of "Roles":
    - Hosted Services (web sites)
    - Worker Services (compute only – pull work from queues)
  - Scalable non-relational storage
    - Dynamic data structure, no static schema
    - SQL Server services added recently
  - Applications implemented using .NET (or PHP)
  - Services can be automatically and seamlessly replicated on multiple physical machines in order to meet configurable performance standards.

Google Cloud Services

- ## AppEngine
  - Designed only for building traditional web apps
  - Develop presentation tier using Java (JSP) or Python.
  - Presentation tier must be stateless.
  - Additional web servers automatically utilized depending on demand.
  - Persistence is via either BigTable or MemCache.

- ## Datastore (Big Table)
  - High Replication Datastore vs. Master/Slave Datastore
  - Highly scalable, but non relational database
  - Can't perform joins.
  - Can create whatever indexes you wish
  - Query result must correspond to a contiguous set of rows from some index.

RSS/Atom

- A publish/subscribe protocol for frequently updated web content.

- XML based document contains publication date(s).

- Subscribed RSS clients regularly check for newly published documents.

- RSS clients can be web based, rich desktop applications or mobile device applications.

- Syndicated web sites can also act as clients.

- More recent rival format Atom is very similar.

    - Both RSS and Atom are in wide use.

- The ASP.NET RSS toolkit includes support for consuming as well as publishing RSS feeds.

Rich Browser Based Interfaces

- HTML Standards Based
  - AJAX
  - HTML 5
- Browser Plug-in Technologies
  - Java Applets
  - Adobe Flash
  - Microsoft Silverlight
  - Java FX

Semester 2 Units

- INB374/INN374 "Enterprise Software Architecture"

What will be on the Final Exam?

. . .