Class announcements

- Thursday:
  - Assignment 4 due
  - Messerschmitt Ch 18 (493-512)
  - Talks
    - Jack O'Neil
    - Kian Fattahi

- For Tuesday 11/25:
  - Database assignment due

Infrastructure acquisition

<table>
<thead>
<tr>
<th>Infrastructure</th>
<th>Build and operate</th>
<th>Build but do not operate</th>
<th>Do not build but operate</th>
<th>Neither</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trend</td>
<td>Outsourced operations</td>
<td>System integrator</td>
<td>Service provider</td>
<td></td>
</tr>
</tbody>
</table>

Stovepipe vs. Integrated Infrastructure

**Stovepipe architecture**

- Separate infrastructure that can support many applications

**Integrated Infrastructure**

- Turnkey Solution
  - Single supplier provides all encompassing solution
  - (complete with infrastructure)

Slides adapted from slides for Understanding Networked Applications
By David G Messerschmitt. Copyright 2000. See copyright notice.
**From stovepipe to layering**

- Data
- Voice
- Video

Application-dependent infrastructure

Many applications

Integrated Infrastructure (Maybe broken into Additional layers.)

Application-independent

---

**Stovepipe vs. Integrated Infrastructure**

- What are some examples of each?
- What are the advantages of each approach?

---

**Vertical Integration vs. Diversification**

- A company is vertically integrated when it makes rather than buys the subsystems in its products.
- A diversified company produces products across different industry segments.

---

**Vertical Integration vs. Diversification**

- Why do customers favor less vertical integration?
  - Prefer competition amongst component suppliers
  - Mix and match components
  - Reduced lock in
- Disadvantages??
  - Customer needs to integrate components from different suppliers.

---

**Vertical Integration vs. Diversification**

- Why do customers favor diversification?
  - Reduce coordination costs by having to deal with fewer suppliers.

---

**General Trend**

- Less Vertical Integration
- More Diversification

- Of course there are exceptions...
Today's supplier structure

- Applications
- Frameworks and components
- Middleware
- Infrastructure (network, OS) software
- Equipment (network, computers)
- Semiconductors, components

Standardization

Purpose of a standard?

- Allow products or services from different suppliers or providers to be interoperable

Scope of a standard

Included:
- interfaces (physical, electrical, information)
- architecture (reference model)
- formats and protocols (FAP)
- compliance tests (or process)

Excluded:
- implementation
- (possibly) extensions

Some issues

Once a standard is set
- becomes possible source of industry lock-in;
  overcoming that standard requires a major (~10x?) advance
- may lock out some innovation

In recognition, some standards evolve
- IETF, CCITT (modems), MPEG
- backward compatibility

Databases

by
David G. Messerschmitt
**Databases**

- Treat data as a separate asset
  - May be shared by multiple applications
- Provide protection and integrity features appropriate to mission-critical data
  - Access control
  - Integrity constraints
  - Persistence
  - etc.

**Two capabilities**

- Aggregation: accessing multiple databases
- Sharing: two or more applications accessing the same databases

**Relational table**

<table>
<thead>
<tr>
<th>Table</th>
<th>Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Address</td>
</tr>
</tbody>
</table>

| Record | Field |

**SQL interface**

- SQL (Structured Query Language)
- Presents single abstract interface to the application logic
  - For manipulating, and extracting data from database
  - Standardized, not vendor specific
- Encapsulates various internal details
  - Data partitioning and replication
  - Host mapping
  - File representation
  - etc.

**Database operations**

- “PROJECT”
- “SELECT”

Each operation results in a new table

**Database Operations**

<table>
<thead>
<tr>
<th>Passengers</th>
<th>Departments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Dept ID</td>
</tr>
<tr>
<td>Alice</td>
<td>1</td>
</tr>
<tr>
<td>Bob</td>
<td>1</td>
</tr>
<tr>
<td>Chris</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Dept ID</th>
<th>Dept Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>Engineering</td>
</tr>
<tr>
<td>Bob</td>
<td>1</td>
<td>Engineering</td>
</tr>
<tr>
<td>Chris</td>
<td>2</td>
<td>Sales</td>
</tr>
</tbody>
</table>
Fields, columns, attributes

Records, rows

• Entries are simple data types or compositions of those types
  • Integer, string, etc.