What is Architecture?

How do you architect a solution?

A system is decomposed into interacting subsystems.

Each subsystem may have a similar internal decomposition.

Three elements of architecture:
- Decomposition
- Functionality
- Responsibility
- Interaction
- Cooperation
System examples

Let's quickly look at some system decomposition examples
- Quick tour of information technology systems

Two-tier client/server

- Local-area network
- Server/Mainframe

Three-tier client/server

- Client
- Enterprise data server
- Application server

System integration

- Architecture
  -> subsystem implementation
  -> system integration
- Bring together subsystems and make them cooperate properly to achieve desired system functionality
  - Always requires testing
  - May require modifications to architecture and/or subsystem implementation

Emergence

- Subsystems are more specialized and simpler functionality
- Higher-level system functionality arises from the interaction of subsystems
- Emergence includes capabilities that arise purely from that interaction (desired or not)
  - e.g. airplane flies, but subsystems can't
**Why system decomposition?**

- Divide and conquer approach to containing complexity
- Reuse
- Consonant with industry structure (unless system is to be supplied by one company)
- Others?

---

**Networked computing infrastructure**

by

David G. Messerschmitt

---

**Layering**

- Elaboration or specialization
- Services
- Existing layers

---

**Example of Layering: networking**

- **Application**
  - Messages
- **Transport**
  - Packets
- **Network**
  - Frames
- **Link**
  - Bits
- **Physical**
  - Signals

---

**Software Layering**

- Application
- Middleware
- Operating System

---

**Operating system functions**

- Graphical user interface (client only)
- Hide details of equipment from the application
- Multitasking
- Resource management
  - Processing, memory, storage, etc
- etc
Middleware Functions

- Capabilities that can be shared by many applications, but that is not part of OS
  - Example: Database Management System (DBMS)
- Hide details of OS from application
  - Java Virtual Machine
- More purposes we’ll talk about later.

What’s a database?

Database
- File with specified structure
  - Example: relational table

A Database

Storage Middleware example: DBMS

- Database Management System (DBMS)
  - Manage Multiple databases
  - Allow multiple applications to access common databases
  - Implement standard data "lookup" (query) functions.

The Internet

by David G. Messerschmitt

Intranet

Private internet

Often connected to Internet

- Firewall creates a protected enclave
What is the Internet?

- An internet is a "network of networks"
- Interconnect standard for LAN's, MAN's, and WAN's
- Internet = the major global internet
- A private internet is called an intranet

Client - Server Computing

Client Server Example

Client Server Example - Layers Revealed

3-Tier Client Server Architecture example
3-Tier Client Server Architecture example

Application Server

Client

What is Bob’s Balance?

Shared data

Web Server

Common Gateway Interchange

Database Management System (DBMS)

Database

Relational Database

<table>
<thead>
<tr>
<th>Customer</th>
<th>Balance</th>
<th>Customer Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>$527</td>
<td>Silver</td>
</tr>
<tr>
<td>Bob</td>
<td>$0.50</td>
<td>Bronze</td>
</tr>
<tr>
<td>Charles</td>
<td>$100000</td>
<td>Gold</td>
</tr>
</tbody>
</table>

DBMS Responsibilities

- *Hide Changes in the Database hardware from the Application*
- *Standard operations on the data, including searches, such as a search is called a query.*
- *Separate Database Management from Applications, so that many applications can access the same data.*
- Security, Integrity, Backup, fault tolerance, etc..

3-Tier Client Server Architecture in General

- Takes inputs from client
- Decides what to be done next
- Processes shared data
- Supports multiple applications with common data
- Protects critical data
- Decouples data administration and application administration

Shared data

3-Tier Client Server Architecture example

Application Server

Client

What is Bob’s Balance?

Shared data

Web Server

Java Servlet

Database Management System (DBMS)

Database

In some implementations Application Logic and Web Server can be put on Different machines.
Peer to Peer

- What is peer to peer good for?

Sun N-tier case

- What does Sun make?
  - Workstations
  - Servers
  - Software

Sun Case

How Successful had Sun been up to 1998?

- Founded in 1982
- Open Standards Workstation
  - Unix Operating System (Solaris)
  - TCP/IP networking
- 1988 – Revenues $1 billion
- 1993 - Market value $3.0 billion
- 1997 - Jumped from 3rd to 1st in Unix Server Market.
How Successful had Sun been up to 1998?
- 1993 - “The network is the computer.”
- 1994 - Internet explodes in popularity

Microsoft mid to late 90s
- Dominated Desktop software
  - Users familiar with Windows, Office, etc.
- NT servers
  - Fine for small intranets, “not industrial strength”

Sun N-Tier Case
- What is Java?
  - Programming Language
  - Portable between computers with different operating systems
  - Easy to write programs in
  - Easier re-use
  - But, programs are slow

What problems did the micro era produce?
- Desktops are expensive to maintain
  - TCO for windows PC $9900!
- Every PC had a lot of software that had to be maintained
  - Office, Windows, etc...
- Small differences, like the order in which software is installed, could make different PCs behave differently!

In the Networking Era
- These “bloated” PCs are networked and termed fat clients.
- But networking of PCs offered the possibility of
  - putting most of the functionality into servers
  - Getting rid of much of the software on the client
  - These clients would be called thin clients.
  - Sun, Oracle, and others saw it as the future.

Hardware for thin clients
- A Network Computer (NC) - a computer with minimal hardware that depends on a network connection to a server to function
  - Be careful not to confuse it with the phrase “networked computer!”
  - Example: Sun’s JavaStation (1996-2000)
  - It is the hardware one would use to implement a thin-client computing model.
Another term from that era...

- A NetPC was a PC introduced by Microsoft and Intel in 1996
  - Same software as a normal PC
  - Did not allow users to install their own software
  - NetPC died out
  - Features of it, and Microsoft's Zero Administration Kit, live on in today's version of Windows.

Microsoft Vision

- Keep "fat-client" model
- Add some features to Windows to reduce administration costs

Sun's Vision

- Thin Client model.
- Application Servers with Applications written in Java.
- NCs could retrieve applications from application server as needed.
- Applications compatible with any NC hardware and OS.
- Applications could be fixed, added, updated at the server level, rather than maintaining each PC.

SUN 3 - Tier

Sun N-tier

Sun N-Tier
**Sun's Performance**

Net Revenue

Net Income

**Today**

- 3-tier model common.
- Sun's version of 4-tier model not-common.
- N-tier model where Webserver and Application Server on separate equipment also common.
- Sun's hardware business not strong.
  - Linux on cheap PCs most common servers
  - Microsoft desktops replacing Sun workstations

**Java**

- Common in Server implementations
  - Example: Java Servlet implementing application logic in a banking application.
- Often used to push simple applets onto client
- Not common
  - For "big" desktop applications
  - Office Suite in Java not popular
- Microsoft is still in business...