C/S – Basic Concepts

Contents:
- 2-tier
  - Gartner Model
  - Winsberg’s Model
- Client/Server Balance
- Example
- 3-tier
- n-tier

The Gartner Model
- Became de facto reference model
- Recognizes 5 possible modes of distribution:
  - distributed presentation
  - remote presentation
  - distributed logic
  - remote data access
  - distributed database
- Assumes a 2-tier model and allocates functionality to client or server

Gartner Group Model

GM: distributed presentation
- Distributed presentation
  - only presentation management function shared between client and server
  - everything else remains on the server
  - screen-scraping (emulation-based) applications
  - GUI placed in front of existing character-based interface
  - first step in migration of legacy applications to a GUI

GM: remote presentation
- Remote presentation
  - presentation manager entirely on client
  - presentation logic, data logic and data manager on server
  - X Window System, Web applications where clients are Web browsers

GM: distributed logic
- Distributed logic:
  - application is split into presentation logic and data logic component
  - all presentation management activities on workstation
  - all data management activities on the server
**Basic Concepts and Notation**

- **Client**: user or program that wants to perform an operation over the system. To support a client, the system needs to have a **presentation layer** through which the user can submit operations and obtain a result.
- **Application logic**: establishes what operations can be performed and how they take place. Enforces business rules and establishes business processes.
- **Resource manager**: deals with storage, indexing, and retrieval of data necessary to support the application logic.

**Winsberg’s Model**

<table>
<thead>
<tr>
<th>Functional Area</th>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface</td>
<td>Presentation Mgmt.</td>
<td>Drives display and graphical layout</td>
</tr>
<tr>
<td></td>
<td>Presentation Logic</td>
<td>Logic for screens, interaction w/users</td>
</tr>
<tr>
<td>Application Logic</td>
<td>Application Logic</td>
<td>Business logic and control flow</td>
</tr>
<tr>
<td>Data (or Resource)</td>
<td>Data Logic</td>
<td>Logical data access, consistency rules</td>
</tr>
<tr>
<td>Management</td>
<td>Database Mgmt.</td>
<td>Storage/retrieval/recovery</td>
</tr>
</tbody>
</table>

**GM: remote data access**

- Remote data access
  - database manager resides on server
  - presentation management and data logic reside on client
  - typical of client/server DBMSs (DB2, Oracle, Informix, etc.)

**GM: Distributed Database**

- Distributed Database
  - portions of the database reside on client
  - portions of the database reside on server
  - DBMS manages communication involved
  - limited implementation of full-fledged DDBMS functionality

**Critique of the Gartner Model**

- Distributed processing is not distributed data
  - first 4 levels describe distributed processing
  - fifth level describes distributed data
  - database distribution is orthogonal to processing distribution and is applicable to all 4 layers
  - distributed databases are transparent to the user, the other 4 layers are not

**Winsberg’s Model – Func’ty**
Client/Server Balance

Example - Fat Client
- Integrity and Consistency
  - Data type verification, ranges, etc. – integrity
    - e.g. Date of birth verification
  - Verify existence of data (according to relationships) – referential integrity
    - Department exists (association relationship)
  - Calculate the next employee number [Autonumber] (read last; add one; return it)
    - e.g. Employee number (unique)

Example - Fat Client
- Business Rules
  - They could be application-specific
  - Business rules scattered in many applications
  - Expressed in form of programming code
- Example:
  - Department Bonus (10% for all programmers)

Example - Fat Server
- Integrity and Consistency
  - Based on Stored Procedures
  - A unique version of the verification process is maintained at the server side
  - Autonumber is a Stored Procedure that controls uniqueness

Example - Fat Server
- Business Rules
  - Stored Procedures & Triggers
  - Business rules are located in one place (the server)
- Example:
  - Department Bonus (10% for all programmers)
  - Employee table (column dept) has a trigger that is executed on update or on insert. The trigger in fact executes a store procedure

2-tier, 3-tier, and Multi-tier
- Two-tier architectures are typical of
  - environments with few clients
  - homogeneous environments
  - closed environments (e.g. DBMS)
- Three-tier architectures are required for
  - scalability to thousands of clients
  - access to heterogeneous data sources
  - maintainability (update software on few app. servers instead of thousands of clients)
2-tier, 3-tier, and Multi-tier

- Multi-tier architectures result when
  - functionality is delegated to specialized servers (communication-, web-, application-, data-server)
  - mobile clients are considered (desktop client could act as server to mobile client)
  - considering distributed object systems in which every server can act as client to another server

Multi-tier: What Is Involved?

The Problem Space of C/S

- Three axes:
  - two-tier vs. three-tier vs. n-tier
  - transactional vs. informational
  - relational vs. object-oriented vs. semi-structured

The Problem Space of C/S

Distribution at the different layers

A game of boxes and arrows

- Each box represents a part of the system.
- Each arrow represents a connection between two parts of the system.
- The more boxes, the more modular the system: more opportunities for distribution and parallelism. This allows encapsulation, component based design, reuse.
- The more boxes, the more arrows: more sessions (connections) need to be maintained, more coordination is necessary. The system becomes more complex to monitor and manage.
- The more boxes, the greater the number of context switches and intermediate steps to go through before one gets to the data. Performance suffers considerably.
- System designers try to balance the capacity of the computers involved and the advantages and disadvantages of multiple layers.
Client/Server Balance

- Separation between client and application server is never clean-cut
  - There's no recipe
  - But experiences!
- All depends on the kind of application
- It also depends on the hardware (client and server)
- Cached data are needed when part of the application logic is located on the client
  - Checking of consistency constraints on client side at data entry time
- Depends on the kind of interaction