



# ICT 2402 Software Engineering

---

## Software Design - 2

# Topics covered

- Software design elements
  - Architectural design
  - Detailed design
  - Database design
  - User interface design

# Generic application architectures

- Application systems are designed to meet an organizational need.
- As businesses have much in common, their application systems also tend to have a common architecture that reflects the application requirements.
- A generic architecture is configured and adapted to create a system that meets specific requirements.

# Design elements

- Architectural design
- Detailed design
- Database design
- User interface design

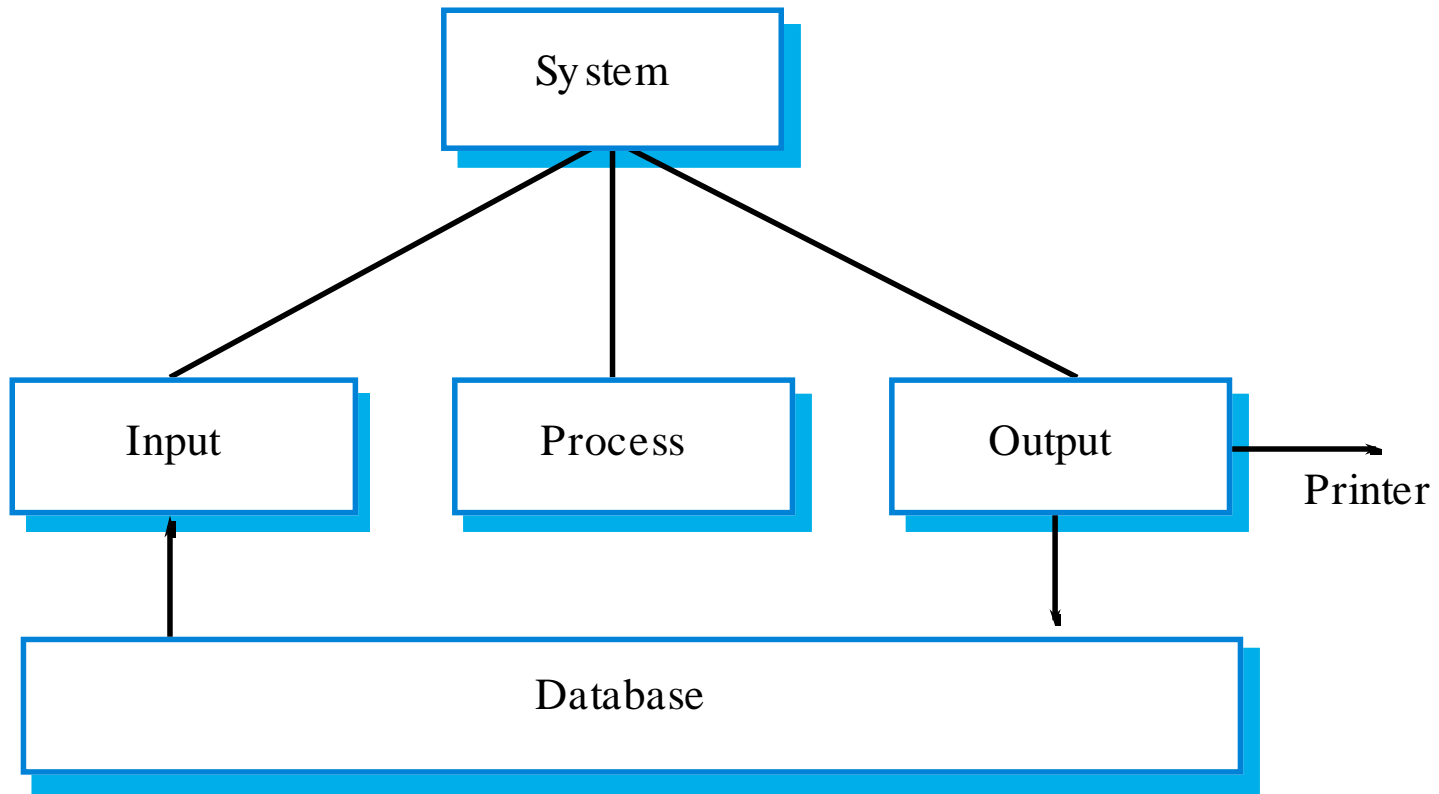
# Application types

- Data processing applications
- Data driven applications that process data in batches without explicit user intervention during the processing.
- Transaction processing applications
- Data-centered applications that process user requests and update information in a system database.
- Event processing systems
- Applications where system actions depend on interpreting events from the system's environment.
- Language processing systems
- Applications where the users' intentions are specified in a formal language that is processed and interpreted by the system.

# Data processing systems

- Systems that are data-centered where the databases used are usually orders of magnitude larger than the software itself.
- Data is input and output in batches
- Input: A set of customer numbers and associated readings of an electricity meter;
- Output: A corresponding set of bills, one for each customer number.
- Data processing systems usually have an input-process-output structure.

# Input-process-output model

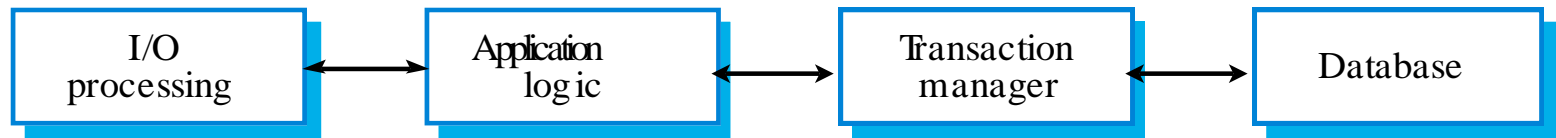


# Input-process-output

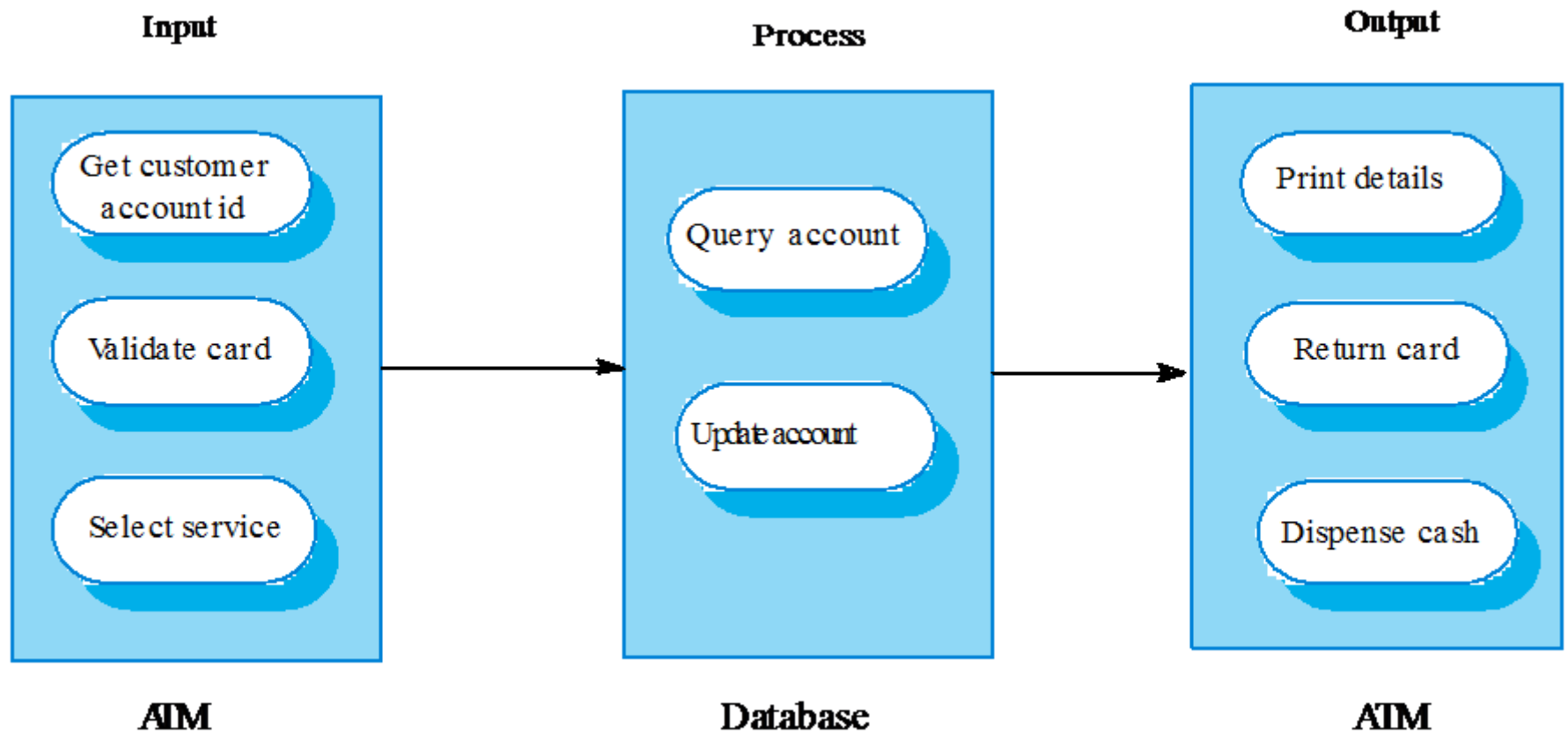
- The input component reads data from a file or database, checks its validity and queues the valid data for processing.
- The process component takes a transaction from the queue (input), performs computations and creates a new record with the results of the computation.
- The output component reads these records, formats them accordingly and writes them to the database or sends them to a printer.



# Transaction processing



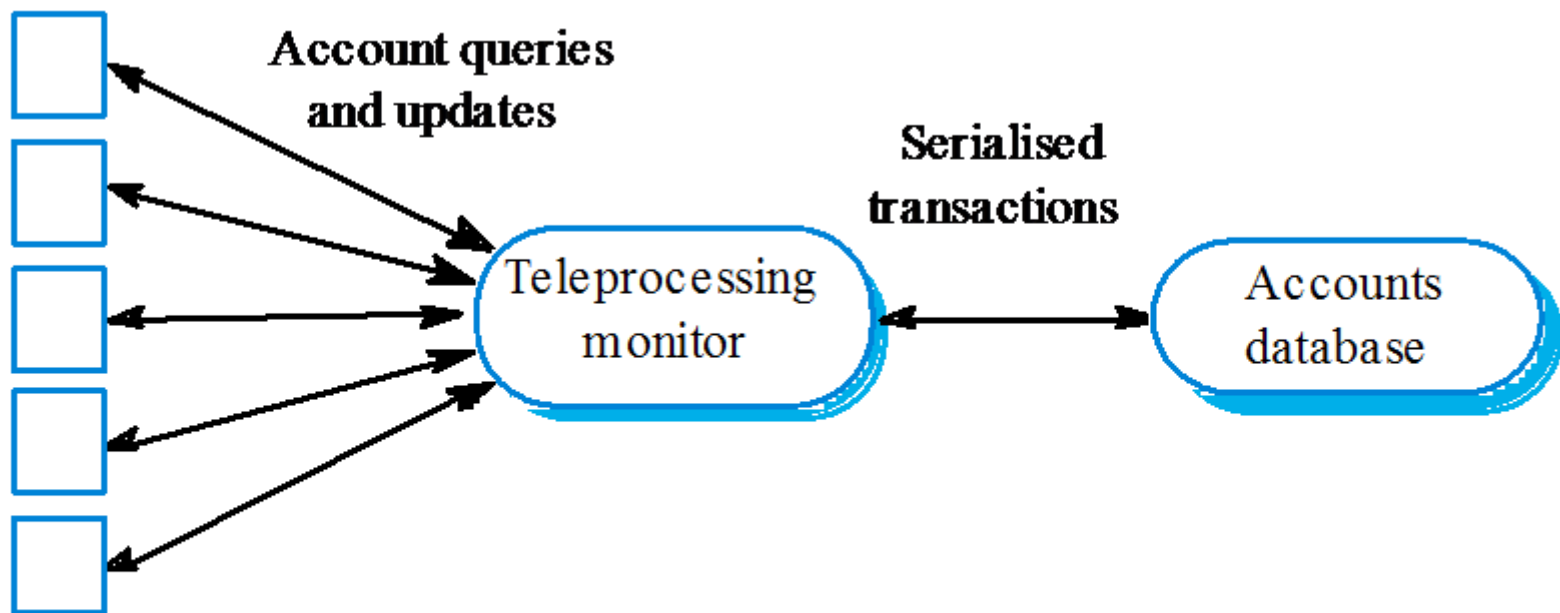
# ATM system organisation



# Transaction processing middleware

- Transaction management middleware or teleprocessing monitors handle communications with different terminal types (e.g. ATMs and counter terminals), serializes data and sends it for processing.
- Query processing takes place in the system database and results are sent back through the transaction manager to the user's terminal.

# Transaction management



**ATMs and terminals**

# Information systems architecture

- Information systems have a generic architecture that can be organized as a layered architecture.
- Layers include:
  - The user interface
  - User communications
  - Information retrieval
  - System database

# Information system structure

User interface

User communications

Information retrieval and modification

Transaction management  
Database

# Resource allocation systems

- Systems that manage a fixed amount of some resource (football game tickets, books in a bookshop, etc.) and allocate this to users.
- Examples of resource allocation systems:
  - Timetabling systems where the resource being allocated is a time period;
  - Library systems where the resource being managed is books and other items for loan;
  - Air traffic control systems where the resource being managed is the airspace.

# Resource allocation architecture

- Resource allocation systems are also layered systems that include:
  - A resource database;
  - A rule set describing how resources are allocated;
  - A resource manager;
  - A resource allocator;
  - User authentication;
  - Query management;
  - Resource delivery component;
  - User interface.



# Layered resource allocation

User interface

User  
authentication

Resource  
delivery

Query  
management

Resource  
management

Resource policy  
control

Resource  
allocation

Transaction management

Resource database

# Layered system implementation

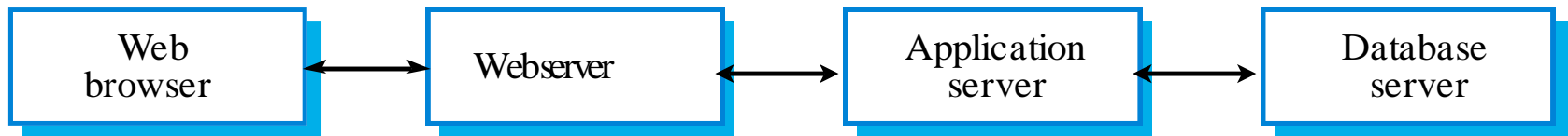
Each layer can be implemented as a large scale component running on a separate server. This is the most commonly used architectural model for web-based systems.

On a single machine, the middle layers are implemented as a separate program that communicates with the database through its API.

Fine-grain components within layers can be implemented as web services.

# E-commerce system architecture

- E-commerce systems are Internet-based resource management systems that accept electronic orders for goods or services.
- They are usually organized using a multi-tier architecture with application layers associated with each tier.



# Event processing systems

- These systems respond to events in the system's environment.
- Their key characteristic is that event timing is unpredictable so the architecture has to be organized to handle this.
- Many common systems such as word processors, games, etc. are event processing systems.

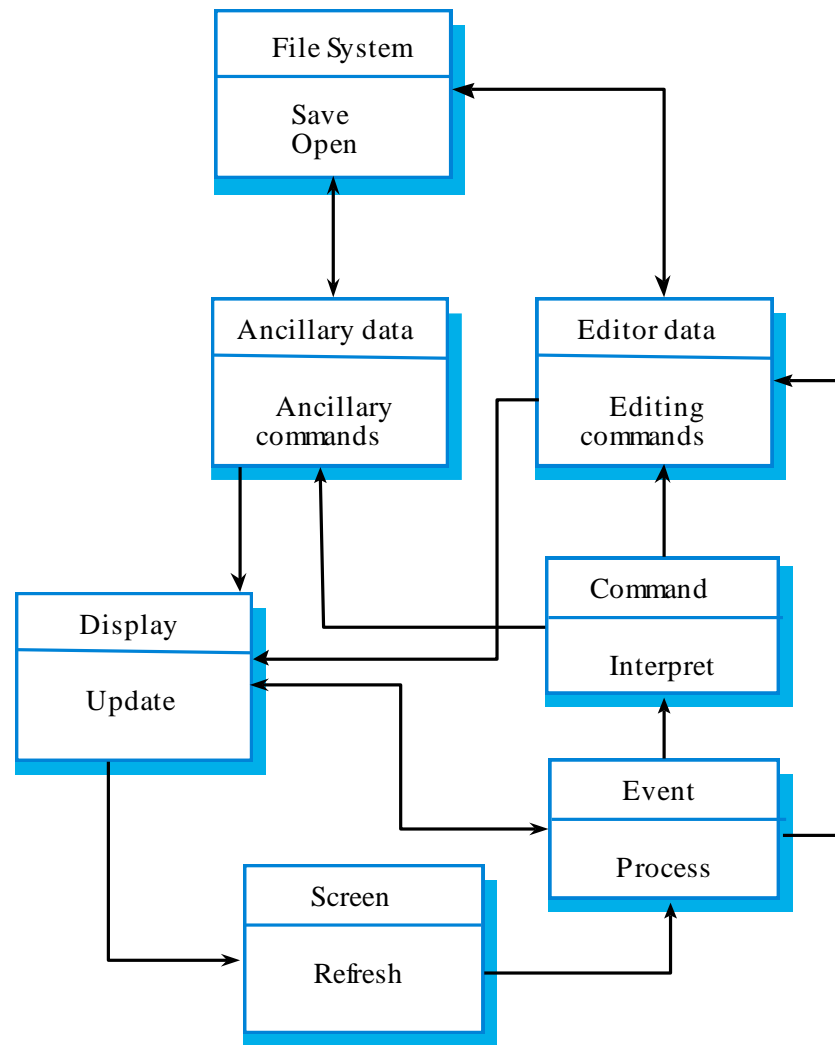
# Editing systems

- Real-time systems and editing systems are the most common types of event processing system.
- Editing system characteristics:
  - Single user systems;
  - Must provide rapid feedback to user actions;
  - Organized around long transactions so may include recovery facilities.

# Editing system components

- Editing systems are naturally object-oriented:
  - Screen - monitors screen memory and detects events;
  - Event - recognizes events and passes them for processing;
  - Command - executes a user command;
  - Editor data - manages the editor data structure;
  - Ancillary data - manages other data such as styles and preferences;
  - File system - manages file I/O;
  - Display - updates the screen display.

# Editing system architecture



# Software design document

Software design document describes how a software system can be implemented starting from the SRS document A typical design document might include,

- Introduction
  - Purpose, scope, assumptions, and reference
- Architectural design
  - High-level hierarchy of modules
- Database design
  - Database schema design
- User interface design
  - Screen design and ordering of screen sequences
- Detailed design
  - Detailed description of each module or method listed in the architectural design



# Design validation and review

- Before moving in to implementation, design should be validated for conformance to the software specification
- Design deliverables may be reviewed by different stakeholders of the system
- According to IEEE standard 1028 on software reviews, 5 types of software design reviews should be conducted.

# Types of design reviews

- Management review
  - A systematic evaluation performed by management to monitor progress, determine status of plans and schedules
- Technical review
  - Evaluate the design by a team of qualified personnel to determine its suitability for its intended use and identify discrepancies from specifications and standards
- Design inspection
  - A systematic peer examination that aims at detecting and identifying software product anomalies

# Types of design reviews(contd.)

- Design walk through
  - A walk-through may be held for the purpose of educating an audience regarding a software product. The major objectives are to,
    - Find anomalies
    - Improve the software product
    - Consider alternative implementations
    - Evaluate conformance to standards and specifications
- Audit
  - The purpose of a software audit is to provide an independent evaluation of conformance of software products and processes to applicable regulations, standards, guidelines, plans, and procedures

# Questions?

