# HUST

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

# SOICT

School of Information and Communication Technology

ONE LOVE. ONE FUTURE.


TRƯỜNG ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

# IT3180 – Introduction to Software Engineering

## 12 – System architecture (2)

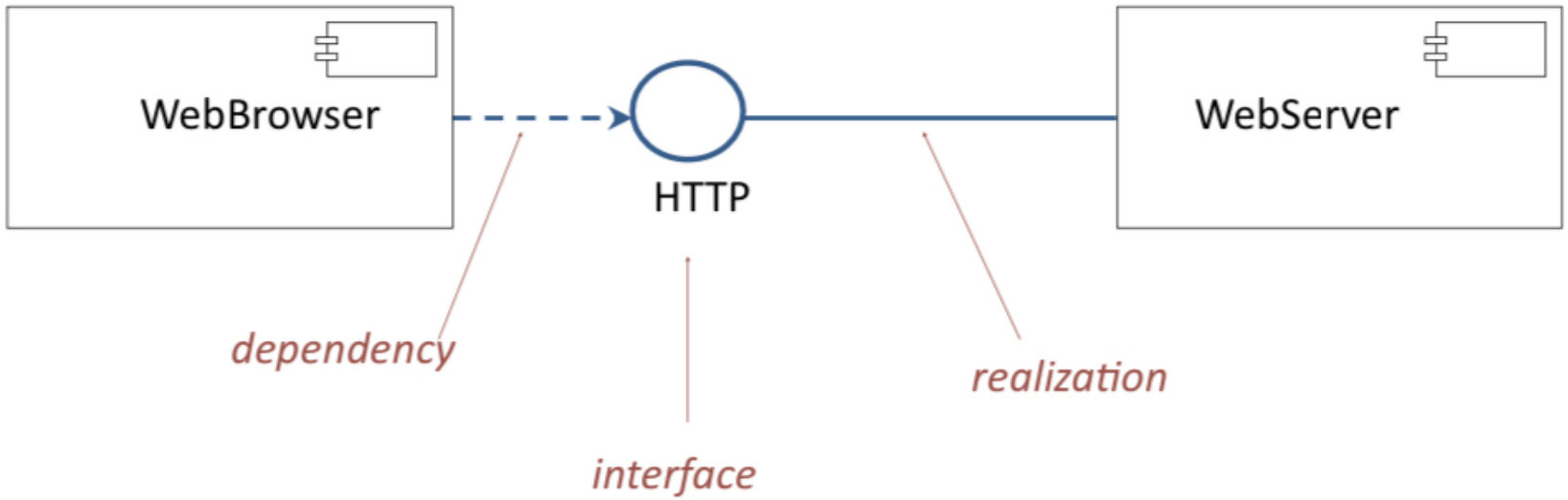
ONE LOVE. ONE FUTURE.

# Three popular architectural styles

- Three tier Architecture
- Master File Update
- Model/View/Control

# Three Tier Architecture

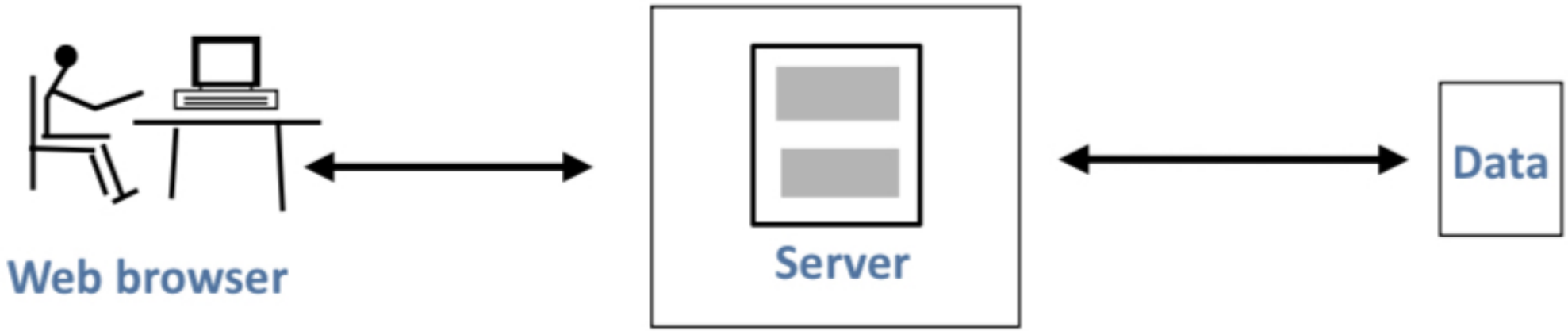
- This architecture is an extension of the client/server model
- It is the standard architecture for small and medium sized web sites
- The basic web system has a client / server architecture



# Web Server with Data Store

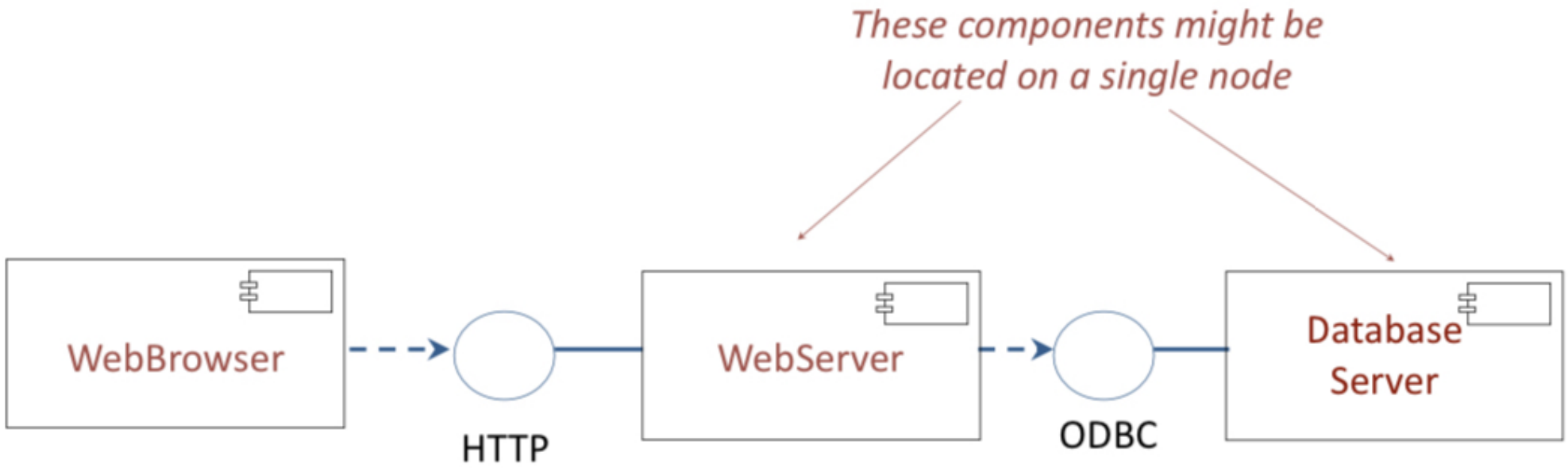
## The basic client/server web site returns only fixed HTML pages

- Attach the server to a data store, so that it can respond to requests from the client and return suitable content



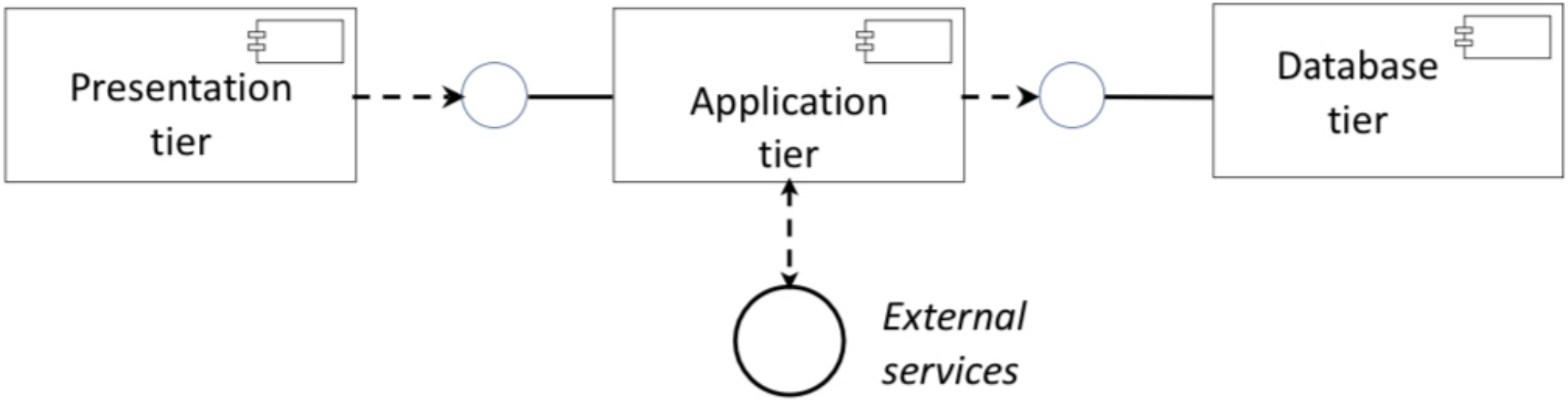

- Advantage: Server-side code can respond to user requests by accessing data, configuring pages, validating information, etc.

# Web Server with Data Store – Component Diagram




- Since components are replaceable binary elements:
  - Any web browser can access the web site
  - The database server can be replaced by another database that supports the same interface

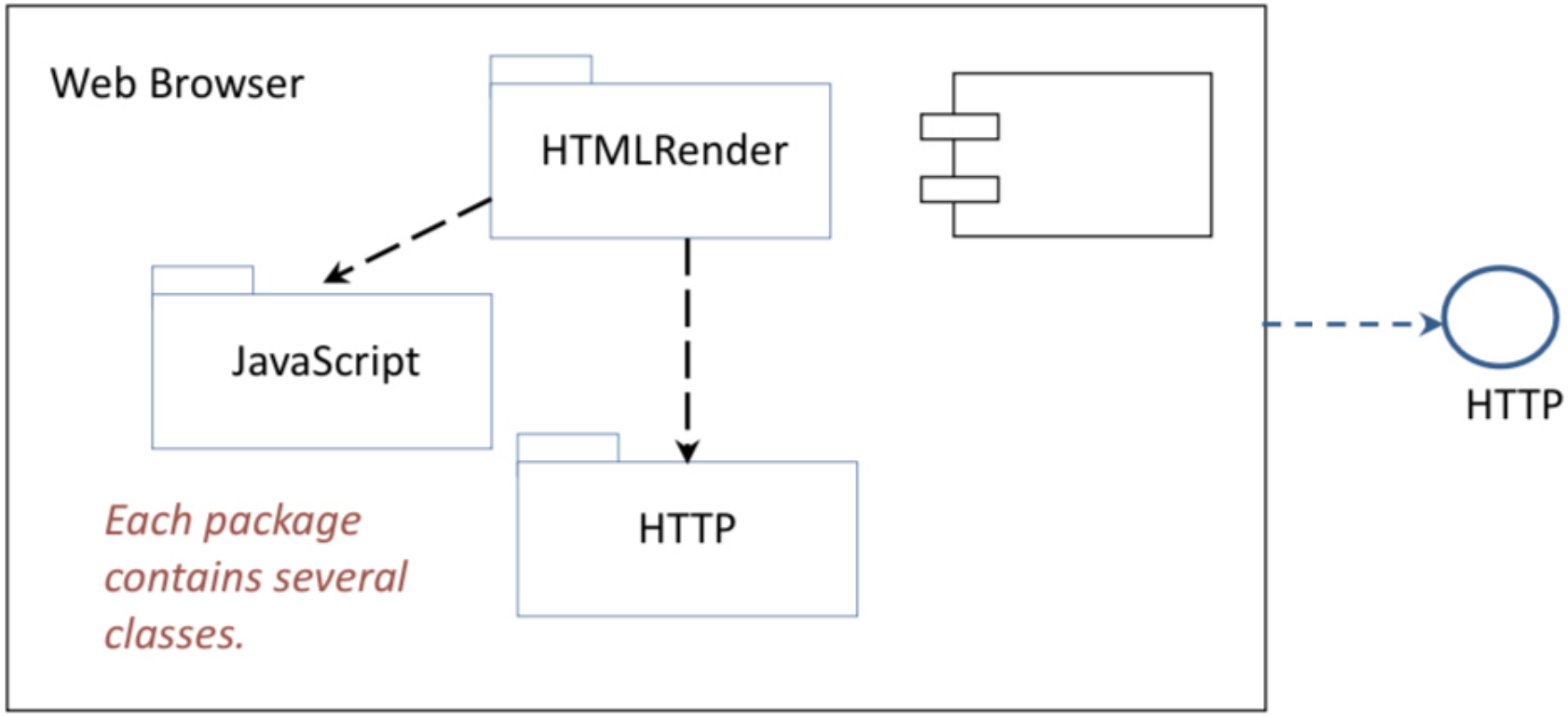
# Architectural Style: Three tier architecture




**Each of the tiers can be replaced by other components that implement the same interfaces**

# Web Browser with JavaScript

## The Presentation Tier has become more complex. Since it still supports the same interface it is still a replaceable binary component.



# Tier vs. Layer

- A ‘layer’ refers to a functional division of the software
  - Presentation Layer
  - Business Logic Layer
- A ‘tier’ refers to a functional division of the software too, but a tier runs on infrastructure separate from the other divisions
- For example:
  - The Contacts app on your phone is a three-layer application but a single-tier application because all three layers run on your phone

# Benefits of three-tier architecture

## A separation of logical functionality and physical functionality

- Each tier can run on a separate operating system and server platform
- Presentation tier, Application tier, Database tier – that best fits its functional requirements
- Each tier runs on at least one dedicated server hardware or virtual server
- Services of each tier can be customized and optimized without impact the other tiers

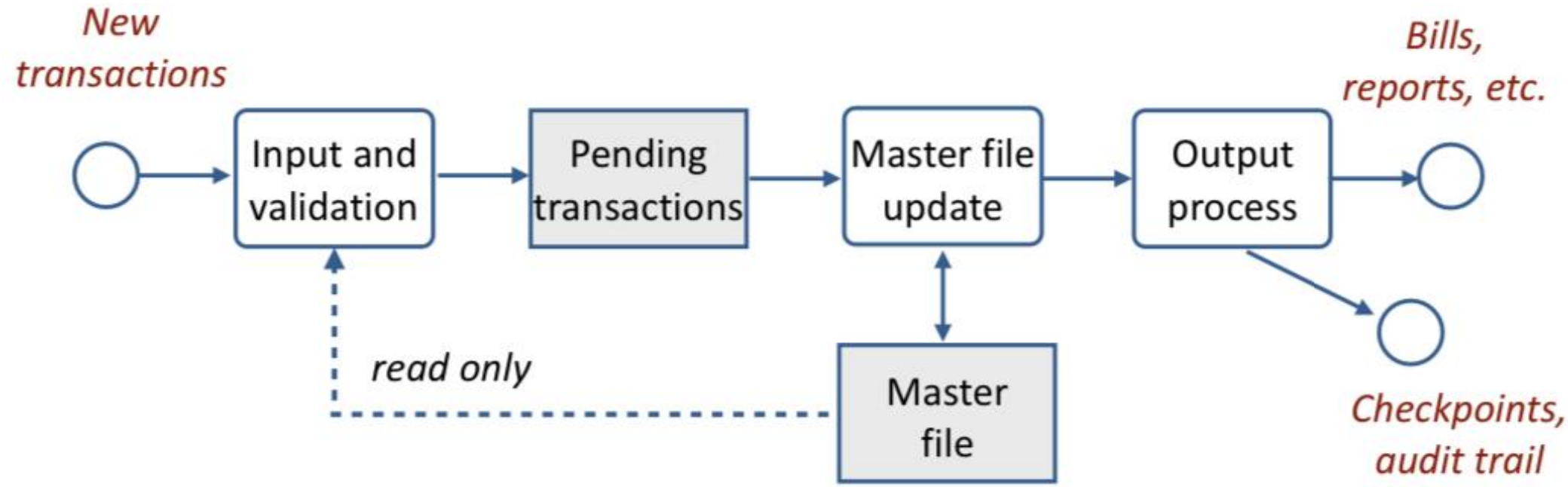
# Master File Update

- This architecture is an alternative to the repository model
- Information is kept on files needing to be modified as changes. The process is called **updating** and the files that are being updated are called **master files**
- It is very widely used in data processing systems
  - Inputs
  - Processing
  - Outputs

# Updating

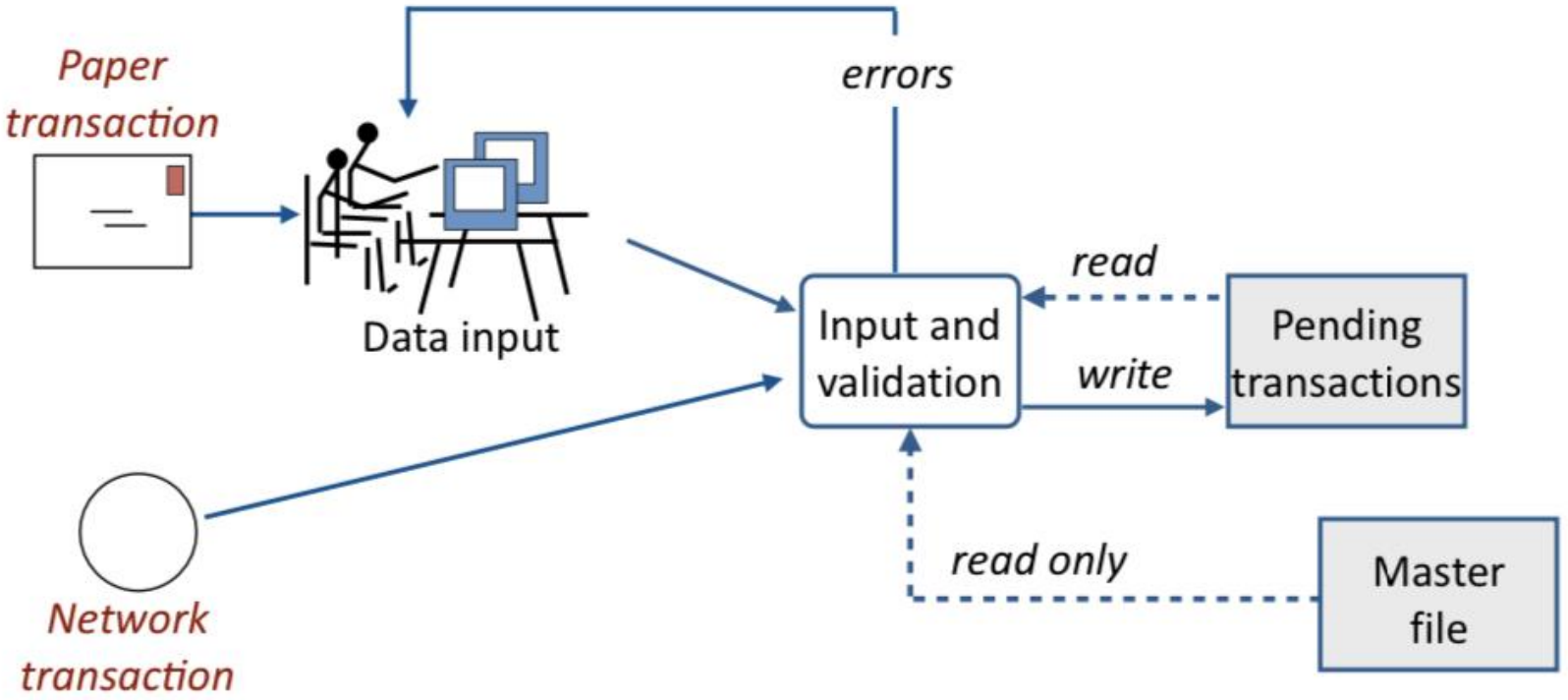
- Updating a file can involve the following:
  - **Adding** records to the file
  - **Changing** records on the file
  - **Deleting** records from the file
- Actions: add, change, delete transactions that will be used to update the master file are saved on a **transaction file**
- Transaction file will be processed with the master file to update the file
- 2 kinds of business updating:
  - **Maintenance** updating
  - **Production** updating

# Master File with Batch Processing: Dataflow Model




- **Input** and **validation process** runs throughout the day
  - It processes transactions when they arrive
- **Master file update** program runs once per day (usually at night)
- **Output process** is run after the master file update finishes

# Batch Processing: Input and Validation




- Because the input and validation process is able to read the master file, it can check that the transaction is compatible with the master file
  - Example: whether or not the file has a record for the customer

# Benefits of Batch processing with Master File Update

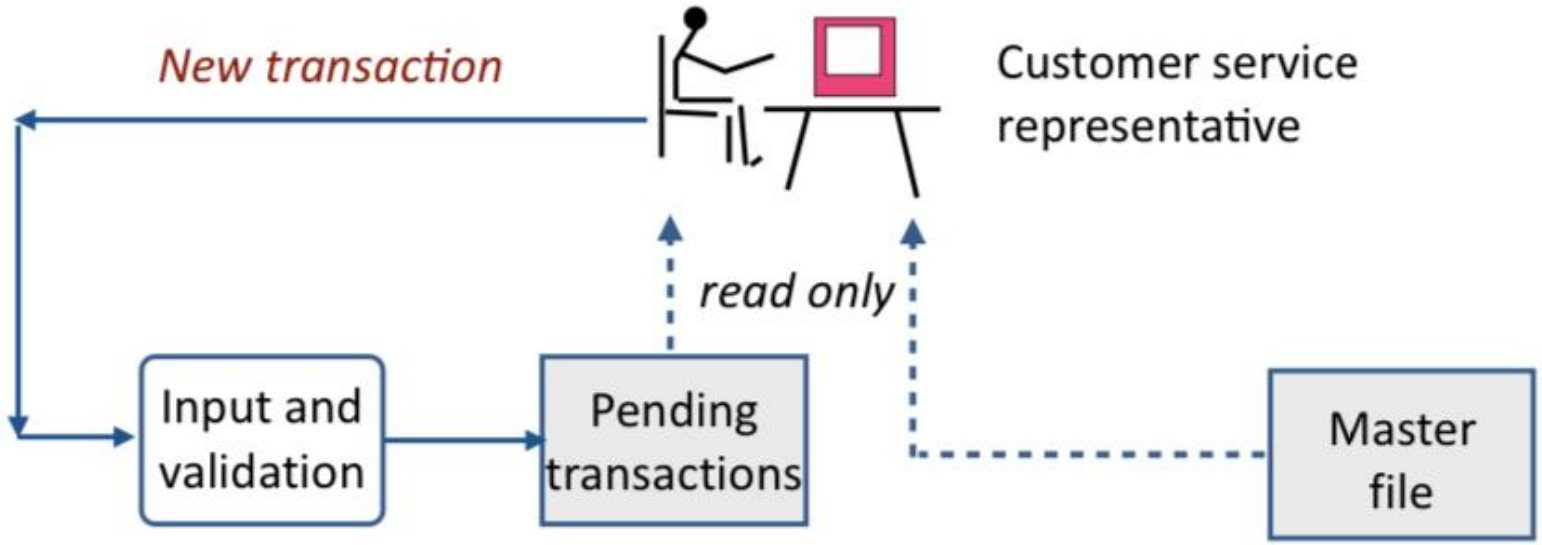
**Advantages:**

- Backup and recovery has fixed checkpoints
- Better management control of operations
- Efficient use of staff and hardware
- Error detection and correction is simplified

**Disadvantages**:

- Information in master file is **not** updated **immediately**
- No good way to answer customer inquiries

# Online Inquiry

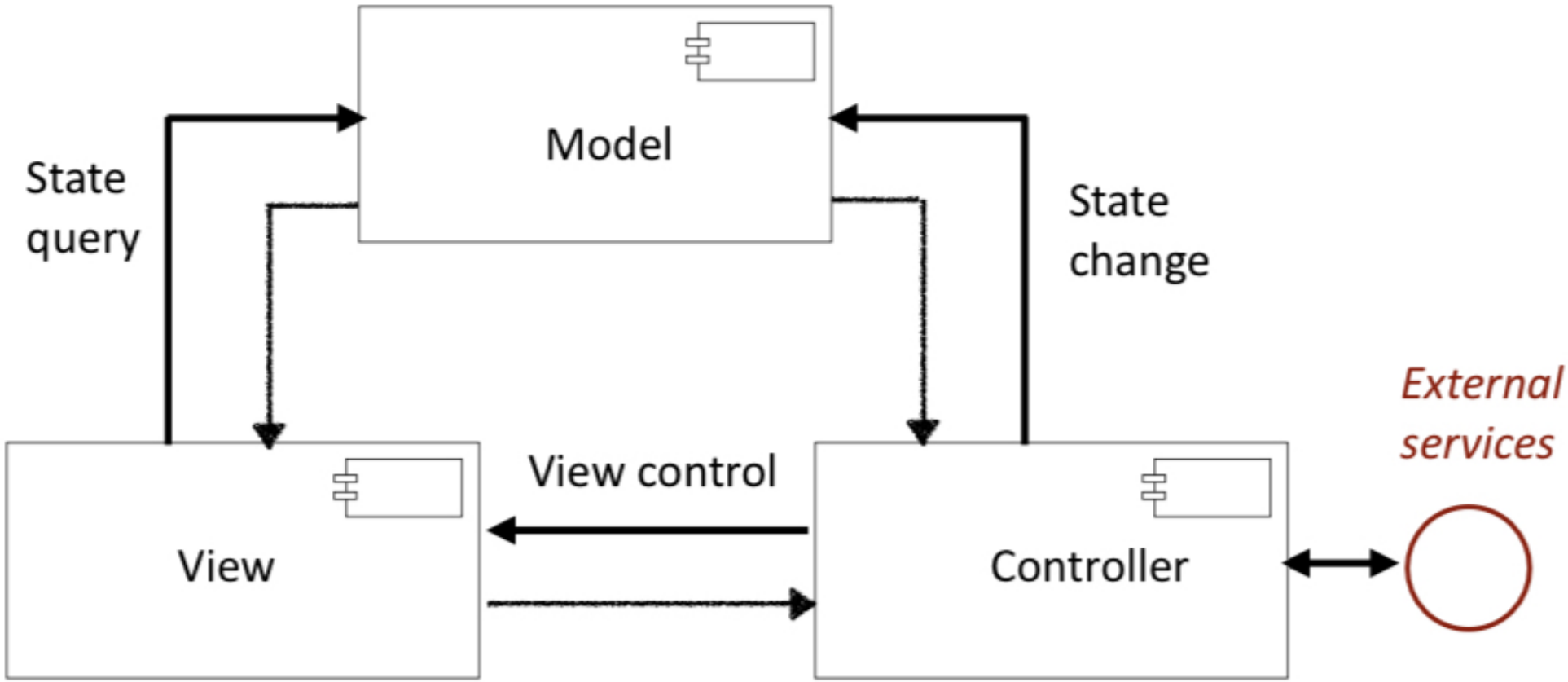



- Customer calls the organization and speaks to a customer service representative
- The representative can read the master file and the pending file, but cannot change them
- If the representative wishes to change information in the master file, a new transaction is created and sent to the input and validation process

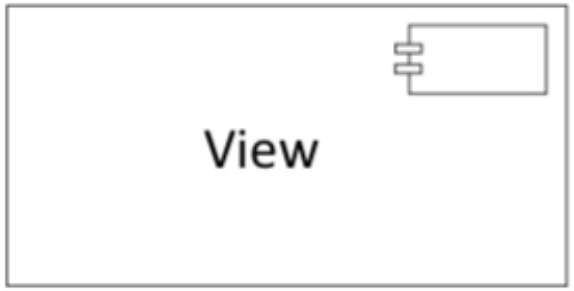
# Model/View/Control (MVC) Architecture

- This architecture is used to control a **complex user interface**
- It is the standard architecture for web/mobile apps and widely used in robotics
- The definition of MVC is in a **state of flux** – The term is used to describe a range of architectures and designs
  - Some are **system architectures**, where the model, view and controller are **separate components**
  - Some are **program design**, with **classes** called model, view and controller

# Model/View/Controller as a System Architecture



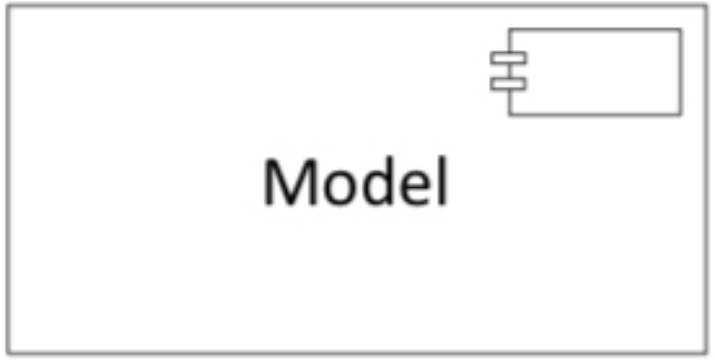
# View




The **view** presents the state of the interface to the user. It subscribes to the model, which notifies it of events that change the state

- Renders data from the model for the user interface
- Provides editors for properties, such as text fields, etc.
- Receives updates from the model
- Sends user input to the controller
- A given model may support a choice of alternative views

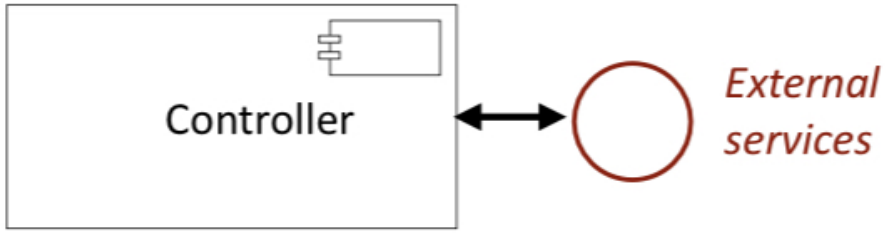
# Model




The model records the state of the application and notifies subscribers. It does not depend on the controller or the view

- Stores the state of the application in suitable data structures or databases
- Responds to instructions to change the state information
- Notifies subscribers of events that change the state
- May be responsible for validation of information

# Controller

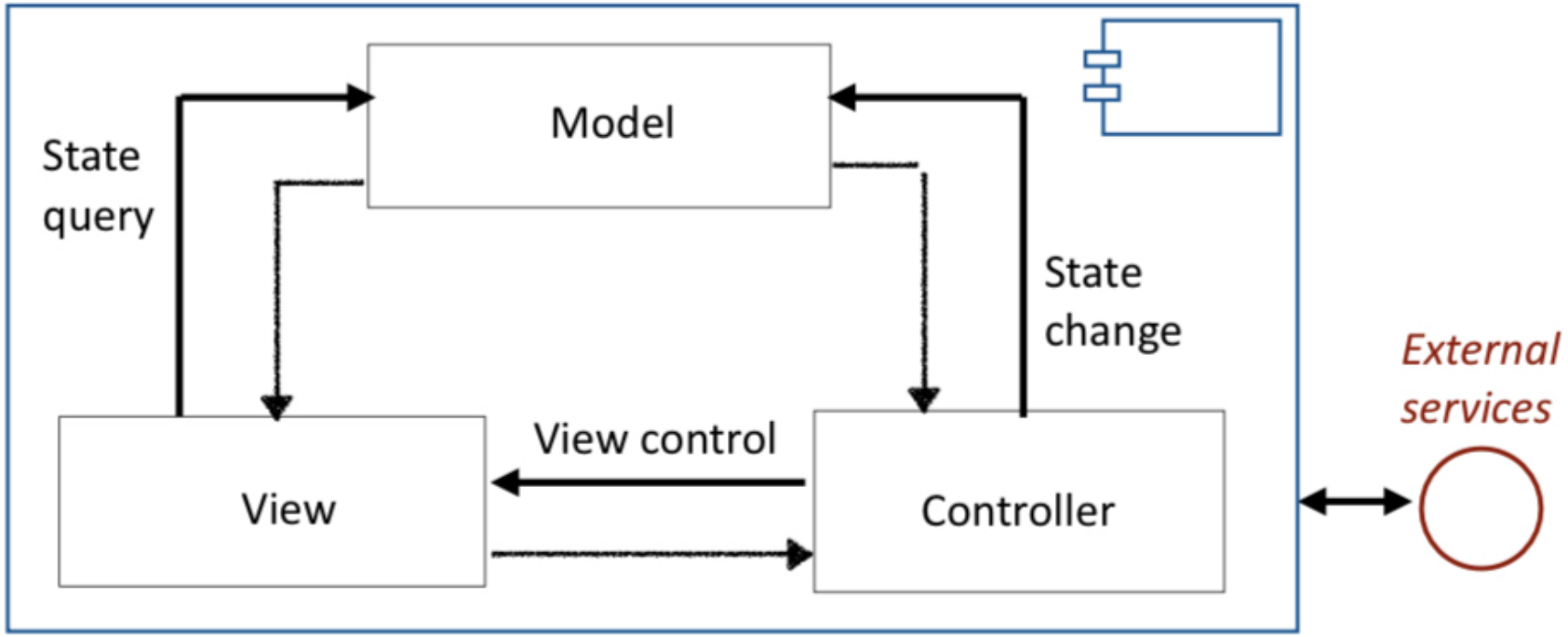



The **controller** is the part of the system that manages user input and navigation within the application

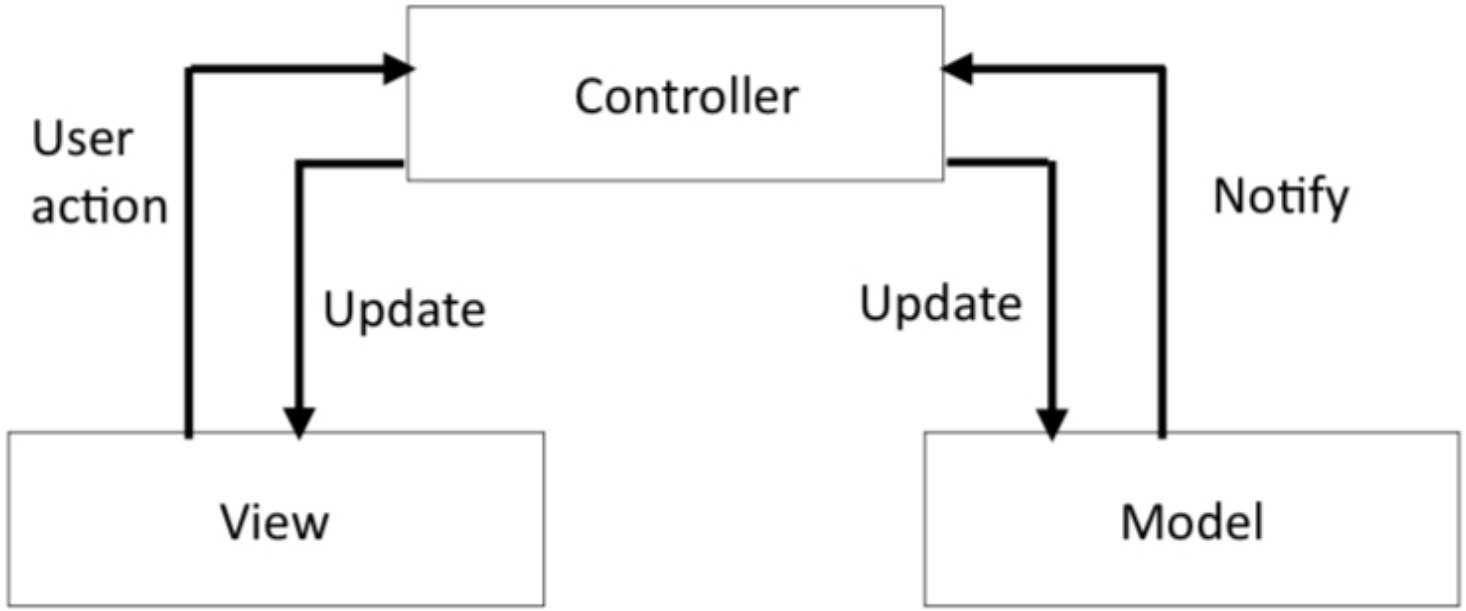
- Defines the application behavior
- Maps user actions to changes in the state of the model
- Interacts with external services via APIs
- May be responsible for validation of information
- Different frameworks handle controllers in different ways. In particular, there are several ways to divide responsibilities between the model and the controller, e.g., data validation, external APIs

# Model/View/Controller as a Program Design

- For mobile apps, the MVC is a single component. The model, view, controller are classes
- The programs often use cloud-based external services, each with an API (e.g., location, validation). These are usually managed by the controller



# Web's Version of Model/View/Controller




- The MVC is a program design (not a system architecture)
- The model, view, controller are classes (not components)
- All messages pass through the controller
- A multi-screen app will have several views and controllers sharing the same model

# Architectural Styles and Design Patterns

- There are many variants of the common architectural styles
- We distinguish carefully between architectural styles and design patterns
  - Architectural styles are part of system design. They are defined in terms of subsystems, components and deployment
  - Design pattern are part of program design. They are defined in terms of classes

# 12. System Architecture (2)

(end of lecture)