		Topics		
Architectural PatternsSections 6.3, 6.4CMPT 276 © Dr. B. Fraser		 What are architectural patterns? What is the advantage of using known application architectures? 		
15-03-06	1	15-03-06		2
Architectural patterns		Model-V	iew-Controller (MVC) pattern	
•		Name	MVC (Model – View – Controller)	
 Object Oriented Design patterns: Common "low level" patterns such as observer and decorator. Used when implementing a system. Architectural pattern: Has been tried and tested in different environments. 		Description Example When used Advantages	 Model: View: Controller: Manages user-interaction. A telephone dialer application: the phone numbers should be stored separate from how they are displayed on the screen. When there are multiple ways to view & interact with the data. Or, when requirements for future interaction are unknown. Supports multiple views of the same data, and changing the data in one view changes it in all views. 	
		Disadvantages	Can add complexity to code if the data (model) and interactions are simple.	
15-03-06 3		15-03-06		4

MVC Diagrams

Controller View selection Maps user actions to model updates Selects view Renders model Requests model updates Sends user events to controller State Change notification	Browser Controller HTTP request processing Application-specific logic Data validation User events Change notification Refresh request	Name Description Example When used	Layered Architecture with related functionality associated with each layer. A layer provides services to the layers above it. A web-based library information system. Different teams developing different layers of the system;
change Model Encapsulates application state Notifies view of state changes	Reitouri request	Advantages	System requires multiple levels of security. Able to replace an entire layer of the system, as long as
The organization of the Model-View-Controller	Web application architecture using the MVC pattern	Disadvantages	Reduces performance due to multiple levels of interpretation, can be difficult to design a clean separation between layers.
15-03-06	5	15-03-06	6
Layered architecture	diagrams	Name	erver architecture Client-server Architecture
Layered architecture o	diagrams Web browser interface		Client-server Architecture
		Name	
User interface User interface	Web browser interface LIBSYS Forms and print login query manager manager Distributed Document Rights	Name Description Example	Client-server Architecture The functionality is organized into Each service delivered by a separate use the servers. (Client and server may be on same computer.) Network file server, web-server (ex: Apache).
User interface User interface management Authentication and authorization Core business logic/application functionality	Web browser interface LIBSYS Forms and query manager Distributed Document Pights	Name Description	Client-server Architecture The functionality is organized into Each service delivered by a separate use the servers. (Client and server may be on same computer.) Network file server, web-server (ex: Apache). When a shared database or service
User interface User interface management Authentication and authorization Core business logic/application functionality System utilities	Web browser interface LIBSYS Forms and query manager Print manager Distributed Document retrieval Rights manager Accounting	Name Description Example	Client-server Architecture The functionality is organized into Each service delivered by a separate use the servers. (Client and server may be on same computer.) Network file server, web-server (ex: Apache).
User interface User interface management Authentication and authorization Core business logic/application functionality System utilities System support (OS, database etc.)	Web browser interface LIBSYS Forms and query manager Print login query manager Manager Manager Distributed Document retrieval Rights Accounting Library index	Name Description Example When used	Client-server Architecture The functionality is organized into Each service delivered by a separateuse the servers. (Client and server may be on same computer.) Network file server, web-server (ex: Apache). When a shared database or service Can replicate servers as required to balance system load. Servers can be distributed over a network,
User interface User interface management Authentication and authorization Core business logic/application functionality System utilities System support (OS, database etc.)	Web browser interface LIBSYS Forms and query manager Distributed Document Rights manager Accounting Library index DB1 DB2 DB3 DB4	Name Description Example When used Advantages	Client-server Architecture The functionality is organized into Each service delivered by a separateuse the servers. (Client and server may be on same computer.) Network file server, web-server (ex: Apache). When a shared database or service Can replicate servers as required to balance system load. Servers can be distributed over a network, Many clients use service without implementing it. Server can be a

Layered Architecture

Client-server example

Client-Server example			
Client 1 Client 2 Client 3 Client 4 Client		Application Architectures	
A cli 15-03-06	ient-server architecture for a film library	15-03-06 10	
Application a	rchitectures	Transaction processing systems	
Generic application architecture: Application Types Description		 Process user requests for information from a database or requests to update the database. Users requests are then processed by 	
Data processing	Data driven app. that without explicit user intervention during the processing.		
Transaction processing	Data-centred applications that process user requests and - Ex: E-commerce systems; Reservation systems.	I/O Application Transaction Database Database The structure of transaction processing applications.	
Event processing	Applications where system actions depend on interpreting events from the system's environment.		
Language processing	Applications where the commands are specified in a formal language that is processed by the system. - Ex: Compilers; Command interpreters.		
15-03-06	11	15-03-06 12	

Web-based information systems

- Web-based Information systems:
 - Transaction-based systems, because these systems generally involve database transactions.

User interface Responsible for all user communications. These systems are often Authentication and implemented as... User communications User interface authorization Implementing application-specific logic Authentication and User communications Information retrieval and modification as well as information storage and authorization retrieval requests; Ex: Shopping cart in an online score. Information retrieval and modification Transaction management (discussed in Chapter 18) Database Transaction management Moves information to and from the Database database and handles transaction management. 13 15-03-06 14 15-03-06 MHC-PMS' architecture Summary Architectural patterns are a means of reusing User interface Web browser knowledge about generic system architectures. - Describes the architecture: Authentication and Form and menu Data User communications Role checking Login validation authorization manager - Explain when it may be used; and - Describe its advantages and disadvantages. Security Patient info. Data import Report Information retrieval and modification generation management manager and export Models of application systems architectures help us Transaction management Transaction management reuse large components. Database Patient database - Web-based information systems are transaction-based systems implemented as Generic web-based layered MHC-PMS specific architecture. architecture. multi-tier client/server architectures. 15-03-06 15 15-03-06 16

Web-based system layers

Implement the user interface.