

Presentation-tier Thinking

Jeff Yu

Mar 18, 2006

Agenda

- Web-based Application Overview
- HTTP Protocol Overview
- Presentation-tier Design consideration
- Some Patterns involved Web Framework
- Explore Struts from pattern's view

Web-based Application Overview

- Client-server model based on HTTP Protocol.
- It is generally form-based application.
- Multi-tier solution:
 - * Persistence tier
 - * Business tier
 - * Presentation tier

HTTP Protocol Overview

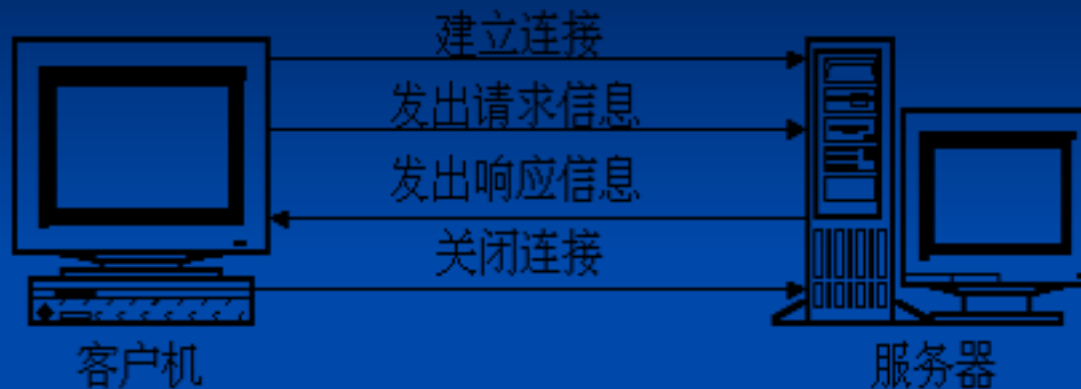
- HTTP stands for Hypertext Transfer Protocol

It's the network protocol used to deliver virtually all files and other data (collectively called resources) on the World Wide Web, whether they're HTML files, image files, query results, or anything else.

- HTTP is based on TCP/IP

- HTTP is called a stateless protocol

HTTP Protocol Overview



- Request (Header ; method: get, post; content)
- Response (Header; content)
- Session (It is user-based)
- Application -> (ServletContext)

Presentation-tier consideration

- Session Management
- Controlling Client Access
- Duplicate form submissions
- Validation

Session Management

■ Session state on the client

- * HTML hidden Fields

- * HTTP Cookies

Cons: These drawbacks are especially apparent when saving large amounts of state. Saving large amounts of state negatively affects performance. Since all view markup now embeds or contains the state, it must traverse the network with each request and response.

■ Session state on the server

It is retrieved using a session ID

Warning: If it is occurred in the cluster environment, you need consider the replica of state ,an alternative is to store session state in either the business tier or the resource tier in this situation.

Controlling Client Access

■ Policy

We should implement the permission check as an independent module, basically, we should not check the permission in our Action.

■ Interceptor

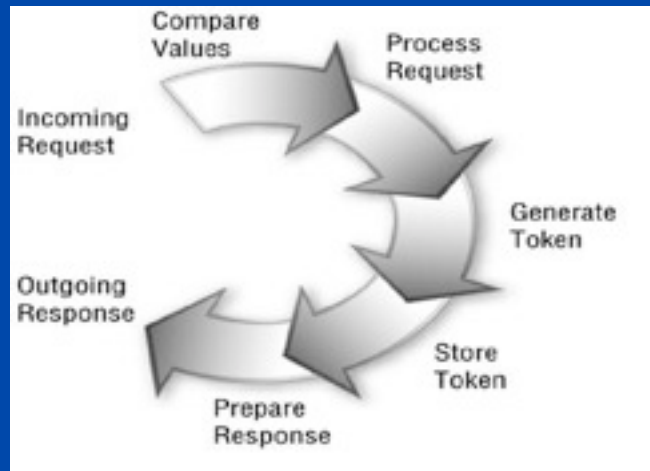
We will weave in the permission module to our system by using interceptor or put the permission module in Front Controller.

Duplicate form submissions

- Hidden buttons via JavaScript

cons: since it used JavaScript, it is not safe, because some users might disable the client script function, and even some people use third-party software to change the JavaScript function and so on.

- Introduce Synchronizer Token



A copy of the token, which should be unique, is stored per user on the server and on the client browser. The token is typically stored on the client browser as a hidden field and on the server in a user session.

Validation

■ Validation on Client

Input validation is performed on the client. Typically, this involves embedding scripting code, such as JavaScript, within the client view. As stated, client-side validation is a fine complement for server-side validation, but should not be used alone.

It also will consider the portable issues among different browsers.

■ Validation on Server

- * Form Centric Validation.

- * Validation based on Abstract Types.

Patterns related Presentation-tier

- Front Controller
- Application Controller
- Command
- MVC

Front Controller

■ Definition

A Controller that handles all requests for a web site.

■ Forces

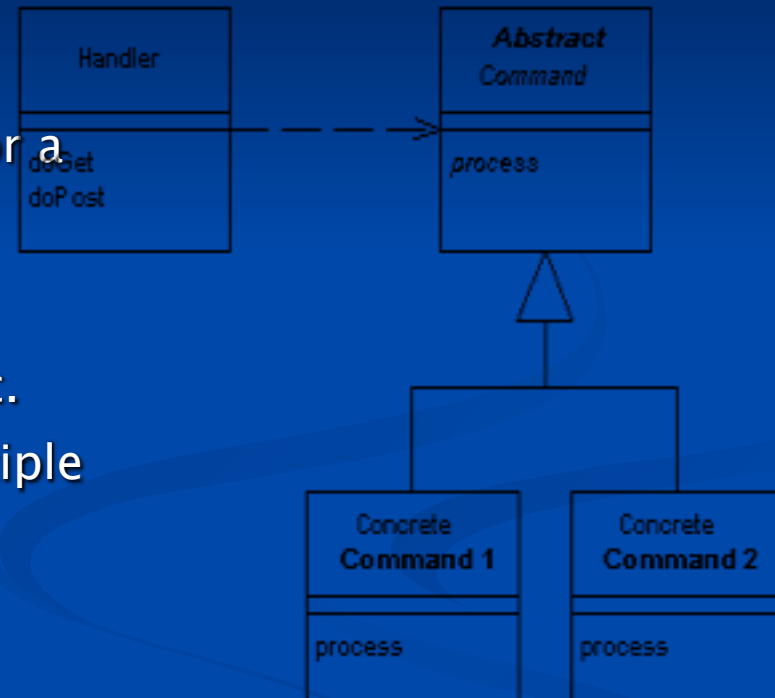
You want to avoid duplicate control logic.

You want to apply common logic to multiple requests.

You want to centralize controlled access points into your system.

■ Responsibility

In a complex Web site there are many similar things you need to do when handling a request. These things include security, internationalization, and providing particular views for certain user.



Application Controller

■ Definition

A centralized point for handling screen navigation and the flow of an application.

■ Forces

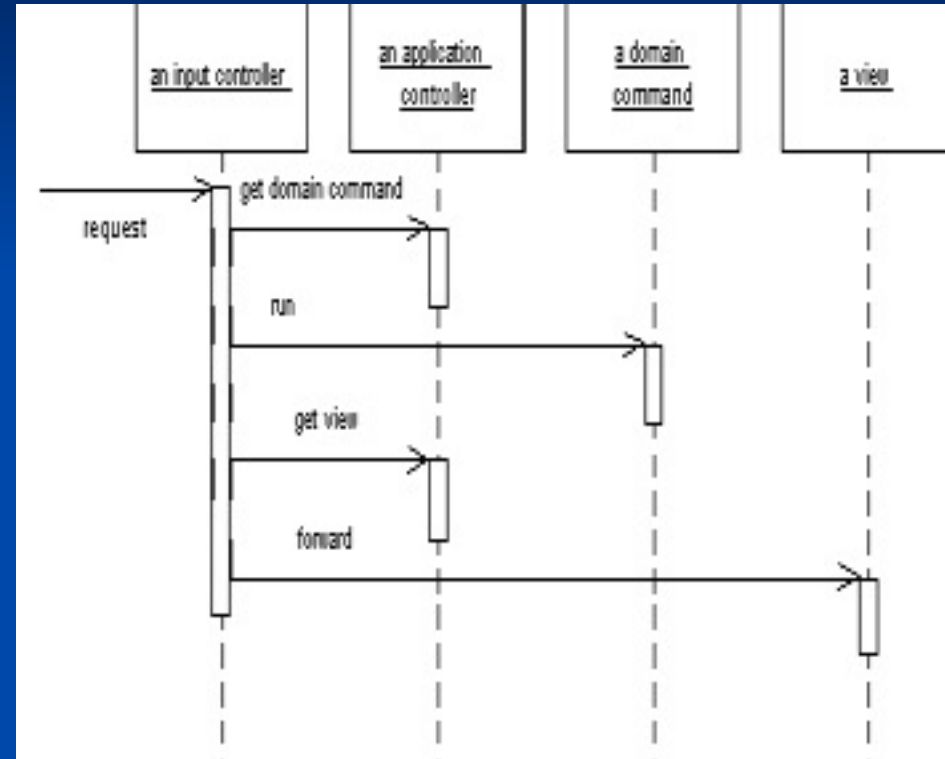
You want to reuse action and view-management code.

You want to improve request-handling extensibility.

You want to improve code modularity and maintainability

■ Responsibility

Deciding which domain logic to run and deciding the view with which display the response.



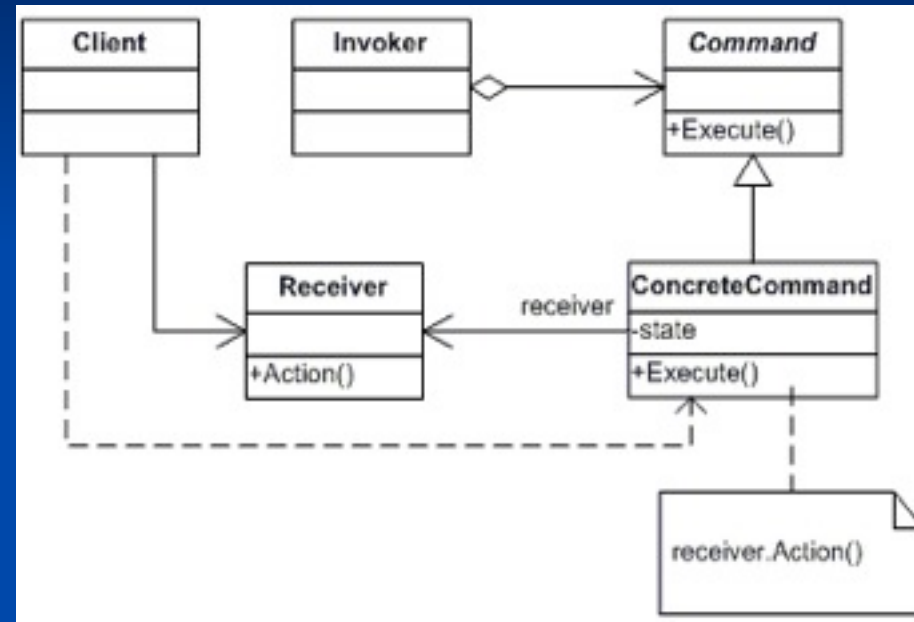
Command Pattern

■ Definition

Encapsulate a request as an object, thereby letting you parameterize other objects with different requests, queue or log requests, and support undoable operations.

■ Benefits

- * It decouple an object making a request from the one that knows how to perform it.
- * A command object is at the center of this decoupling and encapsulates a receiver with an action.



Model View Controller

■ Definition

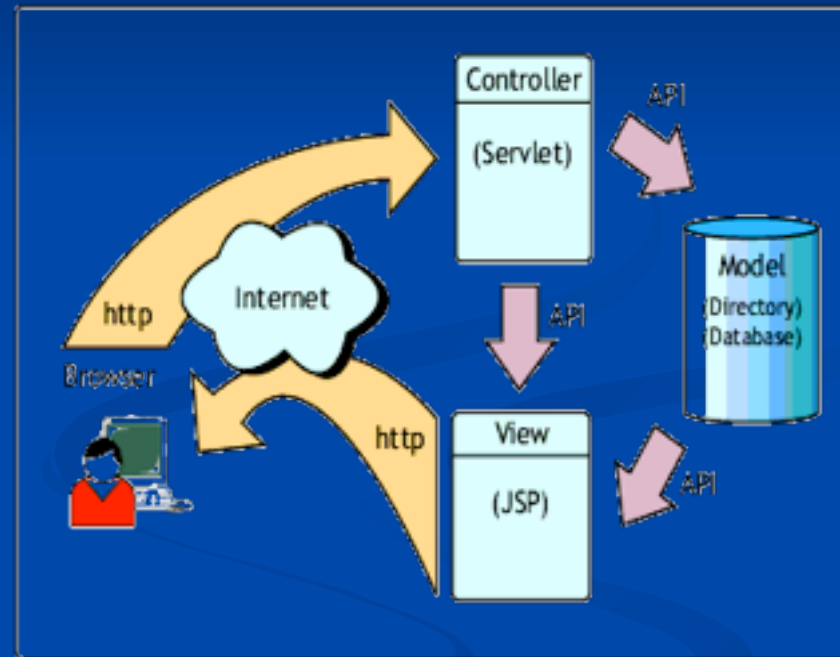
Model: It is an object that represents some information about the domain.

View: It represents the display of the model in the UI.

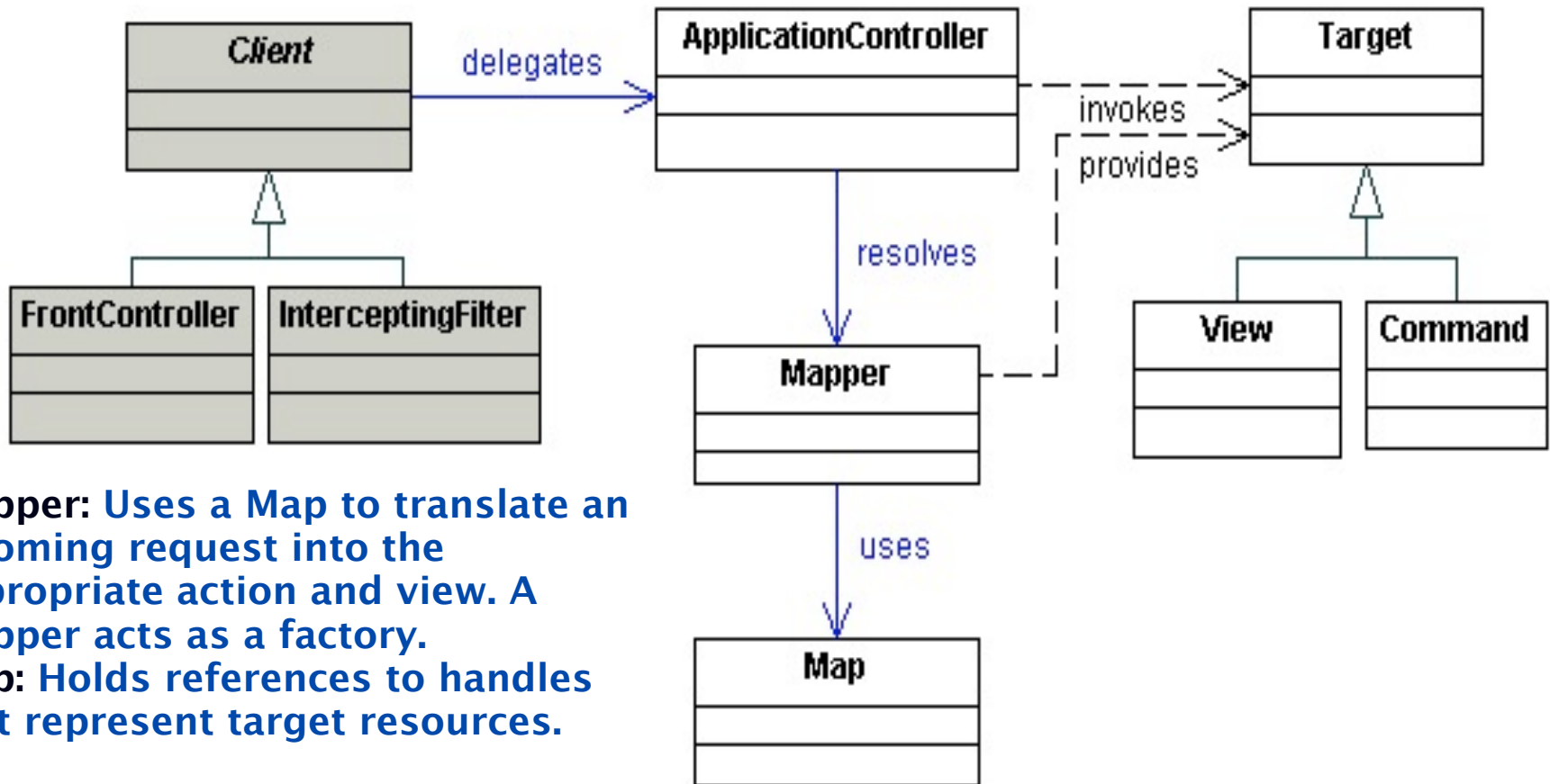
Controller: It takes user input, manipulates the model, and causes the view to update appropriately.

■ Benefits

- * Separating the presentation from the model.
- * Separating the controller from the view.



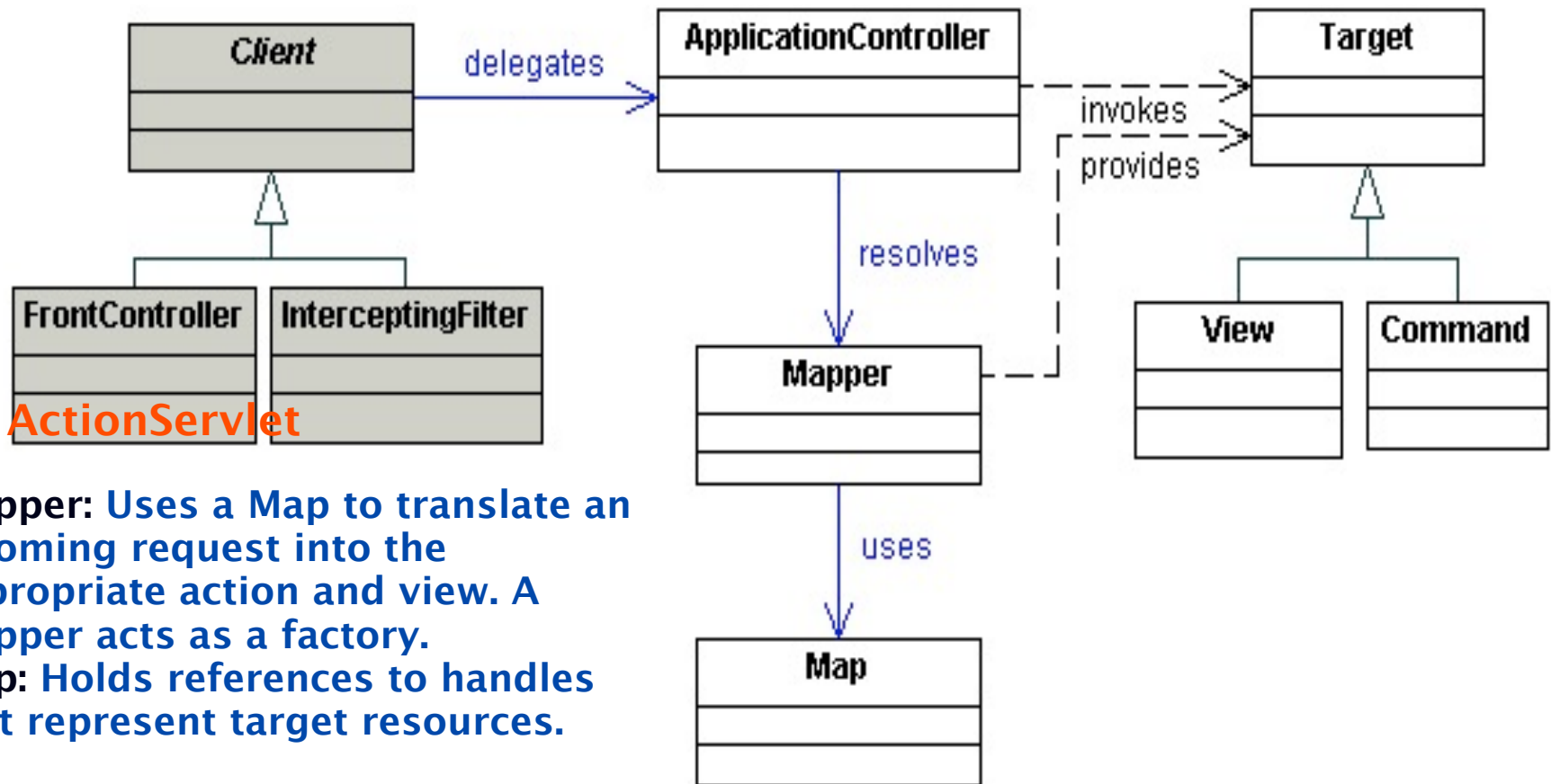
Explore Struts Framework



Mapper: Uses a Map to translate an incoming request into the appropriate action and view. A Mapper acts as a factory.

Map: Holds references to handles that represent target resources.

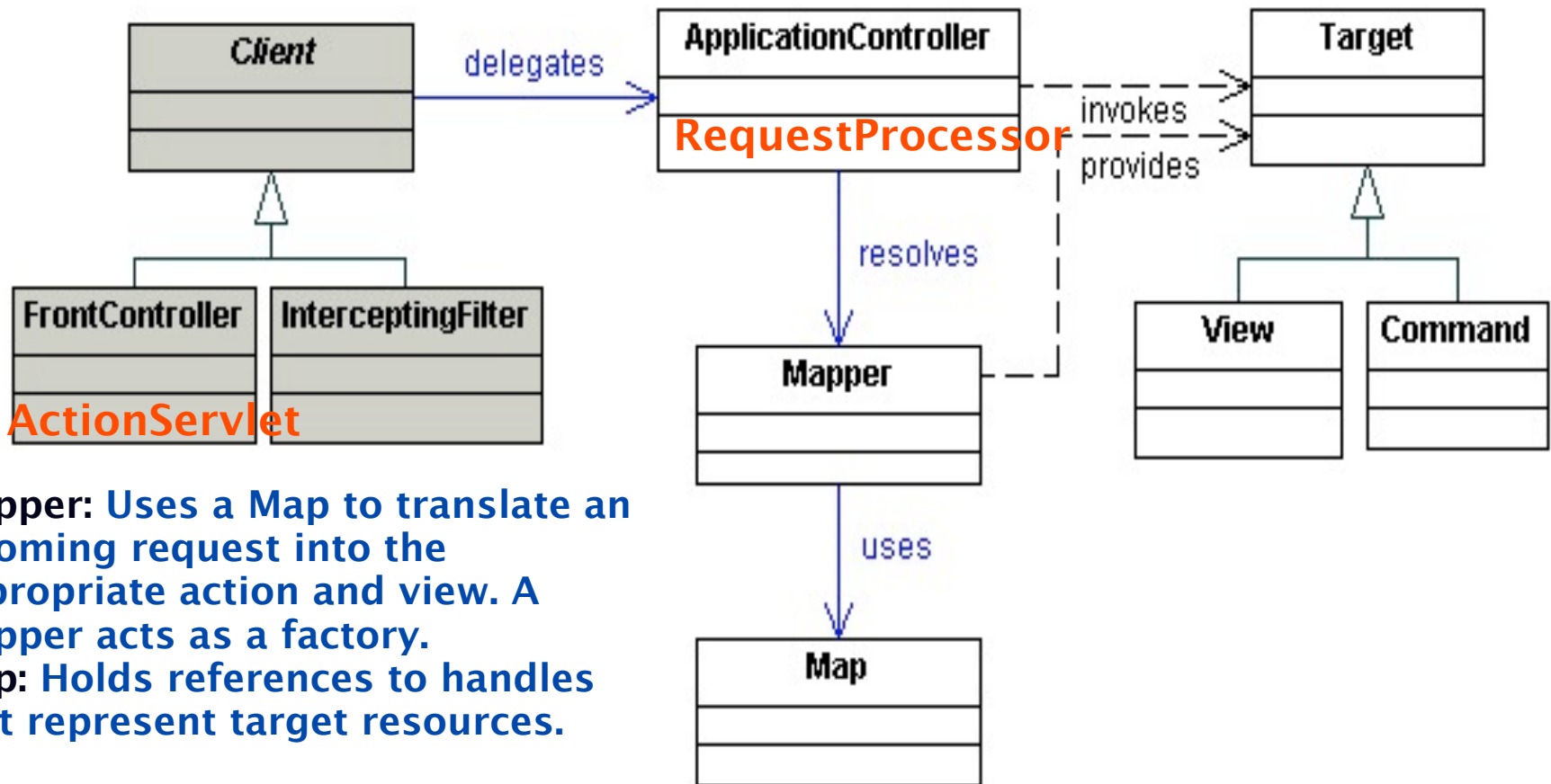
Explore Struts Framework



Mapper: Uses a Map to translate an incoming request into the appropriate action and view. A Mapper acts as a factory.

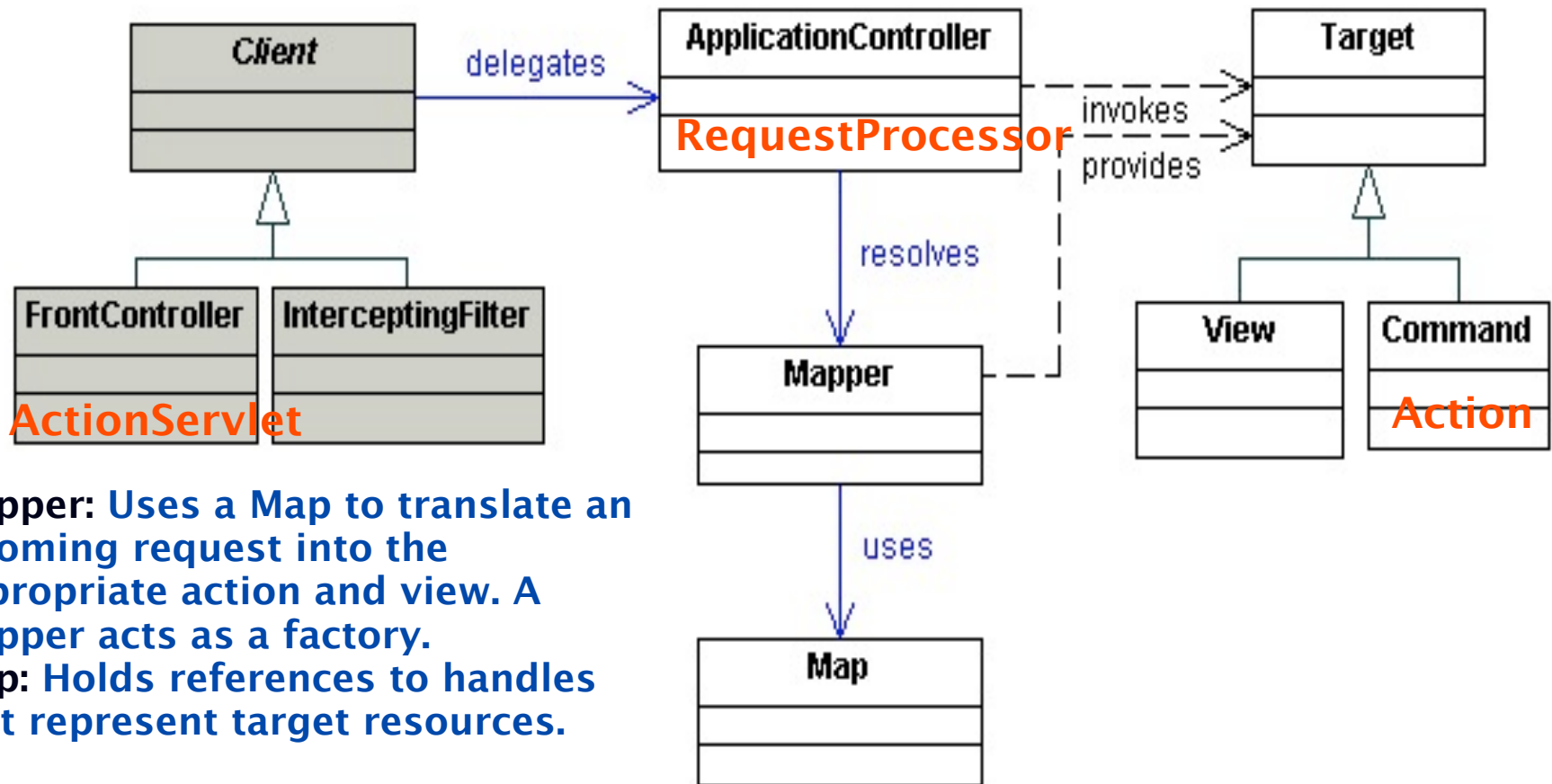
Map: Holds references to handles that represent target resources.

Explore Struts Framework



Mapper: Uses a Map to translate an incoming request into the appropriate action and view. A Mapper acts as a factory.
Map: Holds references to handles that represent target resources.

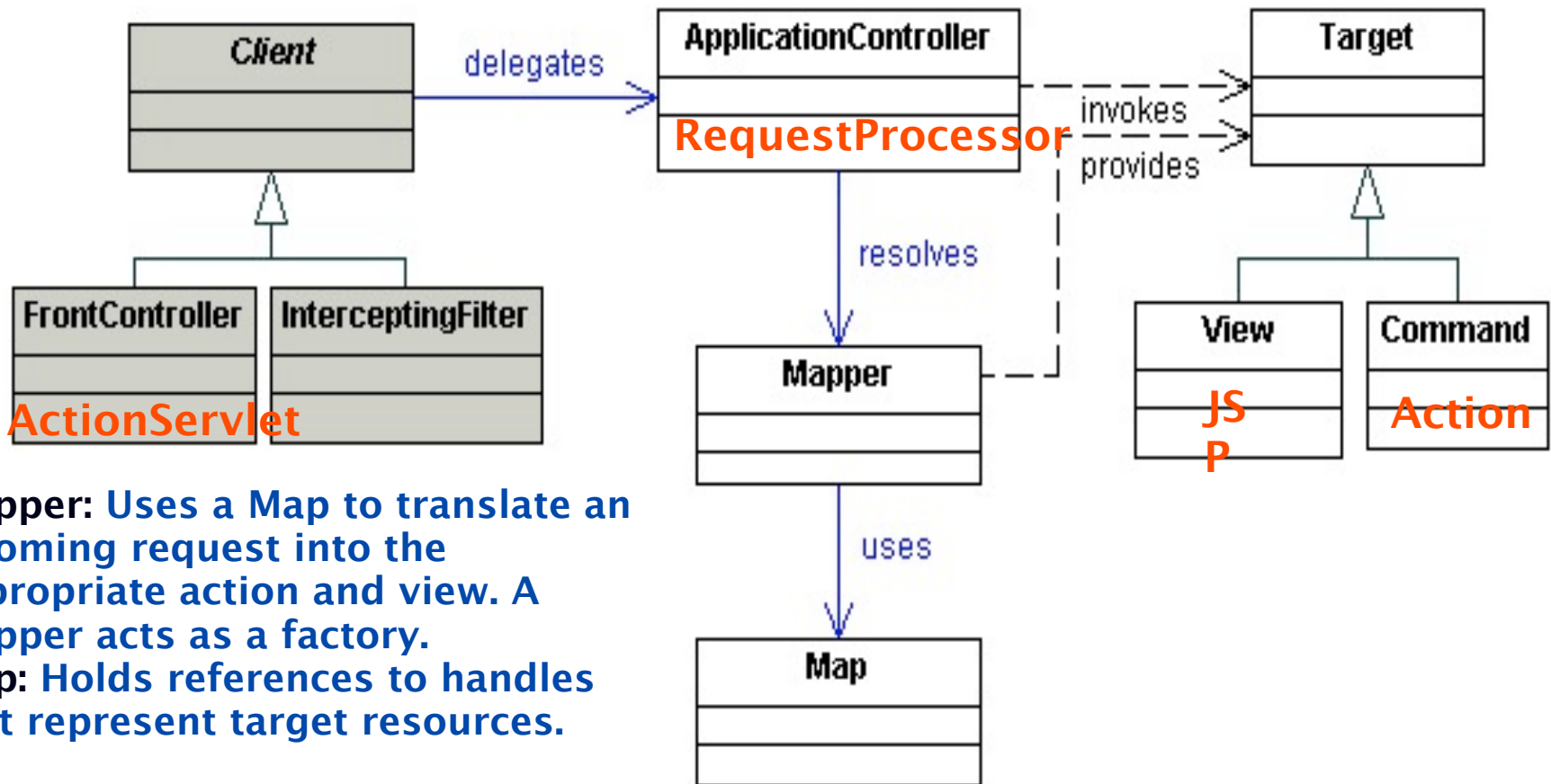
Explore Struts Framework



Mapper: Uses a Map to translate an incoming request into the appropriate action and view. A Mapper acts as a factory.

Map: Holds references to handles that represent target resources.

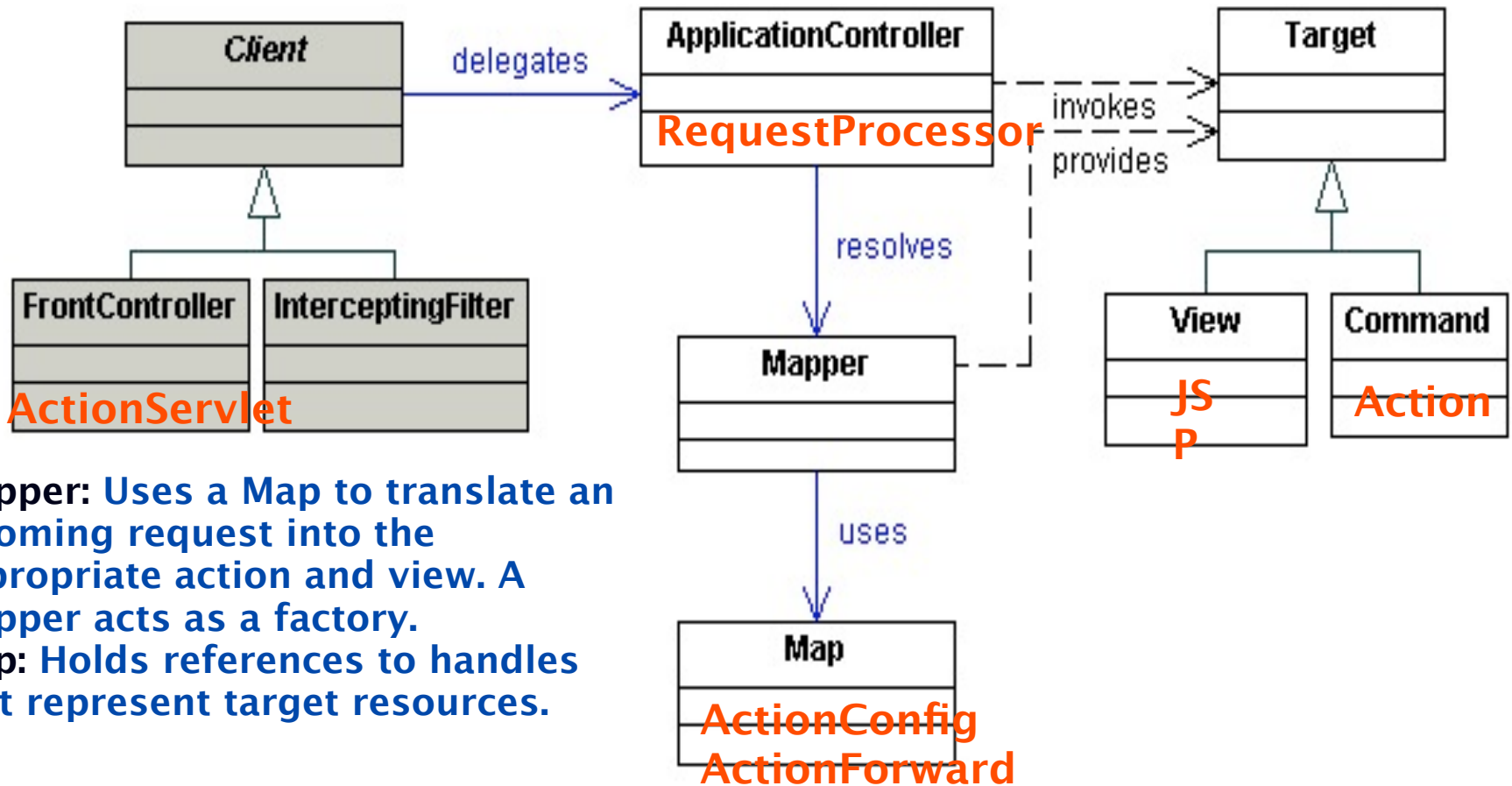
Explore Struts Framework



Mapper: Uses a Map to translate an incoming request into the appropriate action and view. A Mapper acts as a factory.

Map: Holds references to handles that represent target resources.

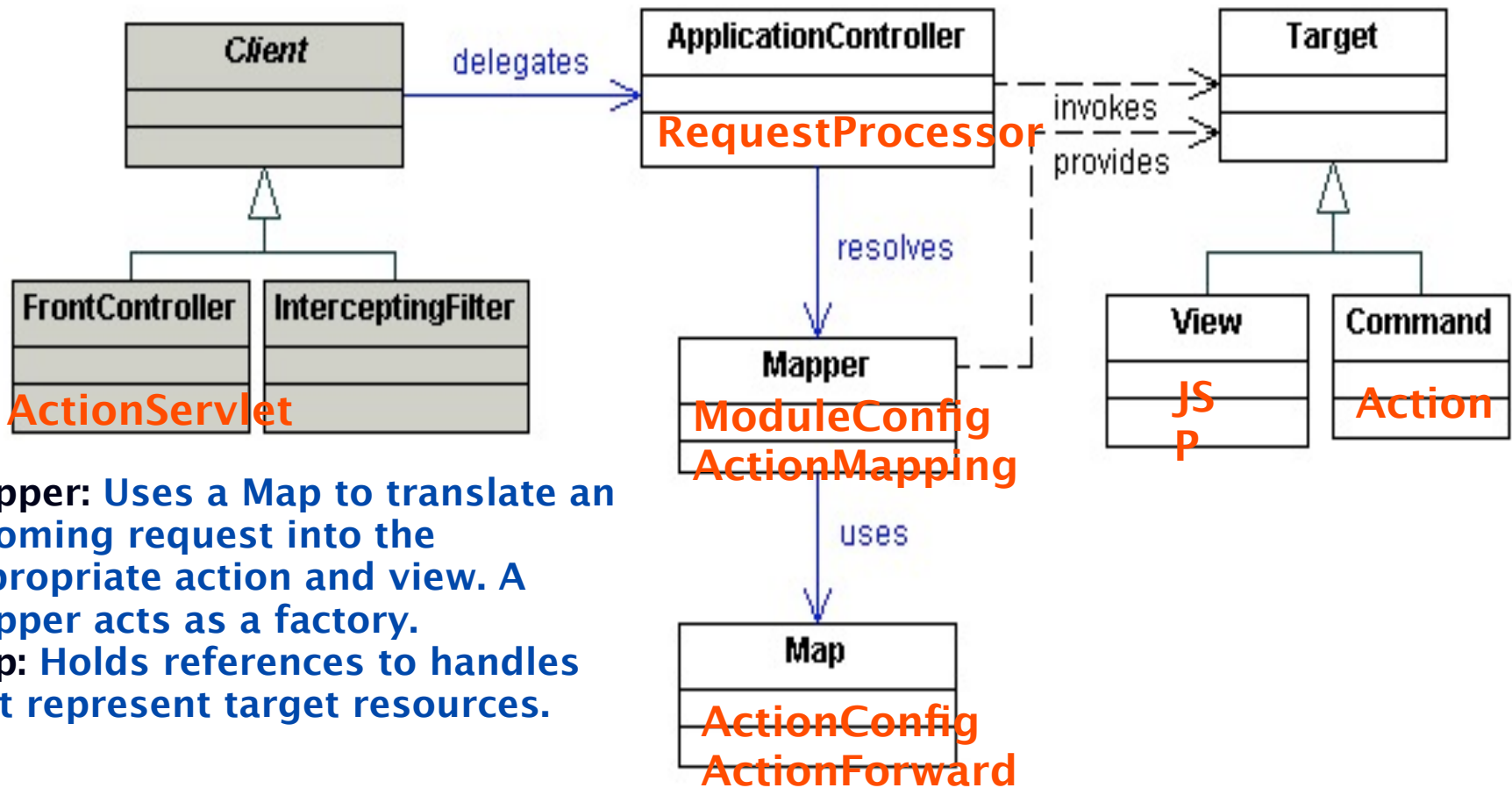
Explore Struts Framework



Mapper: Uses a Map to translate an incoming request into the appropriate action and view. A Mapper acts as a factory.

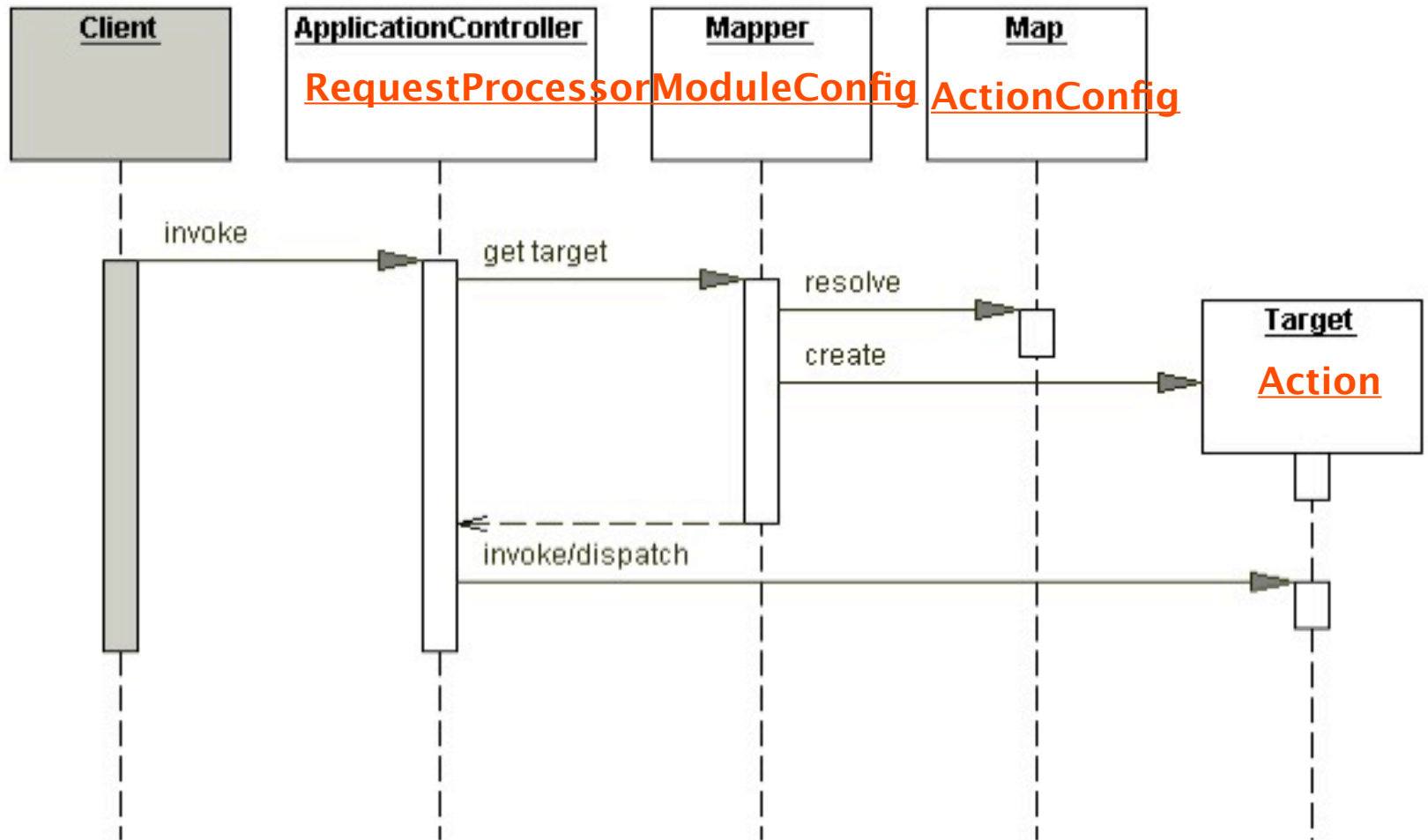
Map: Holds references to handles that represent target resources.

Explore Struts Framework



Mapper: Uses a Map to translate an incoming request into the appropriate action and view. A Mapper acts as a factory.
Map: Holds references to handles that represent target resources.

Sequence Diagram



Q & A

Thanks!!!

Reference:

«Core J2ee Pattern»

«Patterns of Enterprise Application Architecture»

«Head First design patterns»