Introduction

lesson #intro01

James L. Parry
B.C. Institute of Technology

Introduction

This course will help you learn how to develop webapps using CodeIgniter.

It presents a comprehensive set of skills, but not all of CI!

It shows "best practices", but not the only way to use the framework.

The course assumes you know O-O programming and PHP, and that you have some familiarity with HTML.

What Makes a Good Webapp?

"Good" content:

- Informative
- Interesting
- Exclusive

"Good" properties:

- Dynamic
- Personalized
- Scaleable

How Does a Developer Make a Good Webapp?

- Frameworks & Object Models
  - HTTP & web server
  - Distributed systems & deployment
  - Server-side & client-side logic
- Conventions
  - HTML, XML, CSS
- Tools for convenience & productivity
  - IDE (NetBeans)
  - XAMPP
  - Unit testing, RDBMS
  - Github, gitflow, CI -> devops
- Software ... MVC pattern-driven

Websites

Webpages (HTML), intended for browser.

Typified by:

- HTML for page structure
- CSS for layout
- Graphics for aesthetics
- Javascript for client-side logic
- Support technologies for high availability, fast response, analytics

Webapps

To "website", add...

- Usecase driven
- Business logic
- Data driven, dynamic
- Internationalization & localization
- Authentication & personalization
- Support technologies for scalability, management, integration

DevOps

The current trend in software development process!

- Software development methodology
- Communication, collaboration & integration
- Developers & IT staff
- Highly automated, often cloud-based
- Targets product delivery, quality assurance, feature development, maintenance releases, and issue tracking
- Big on agile management!

Technical Terminology

Glue:

A generic term for any interface logic or protocol that connect two component blocks.

Standard Glue (no choice)

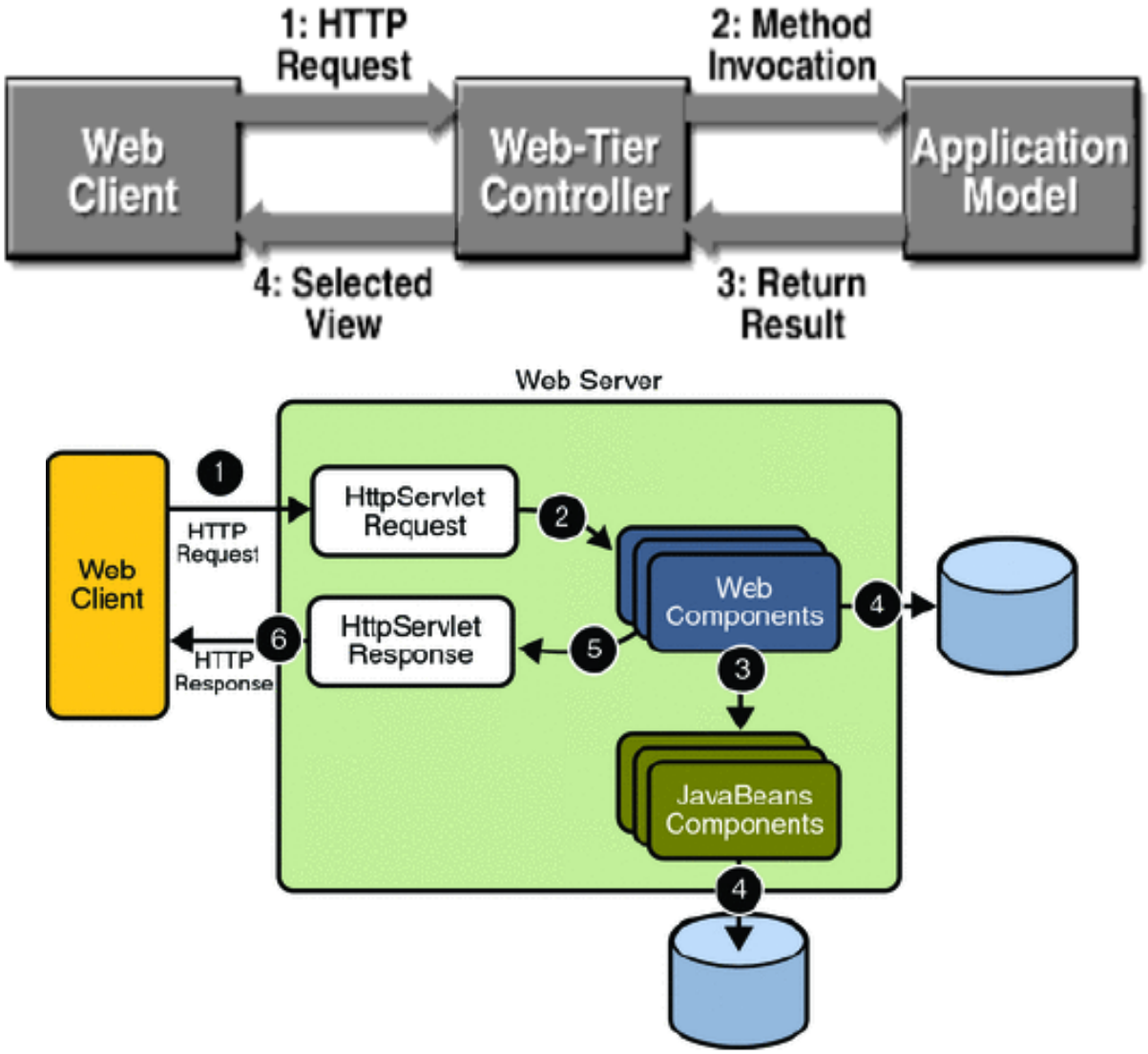
Transport: HTTP, HTTPS, AJAX

Representation: XML, JSON, RSS

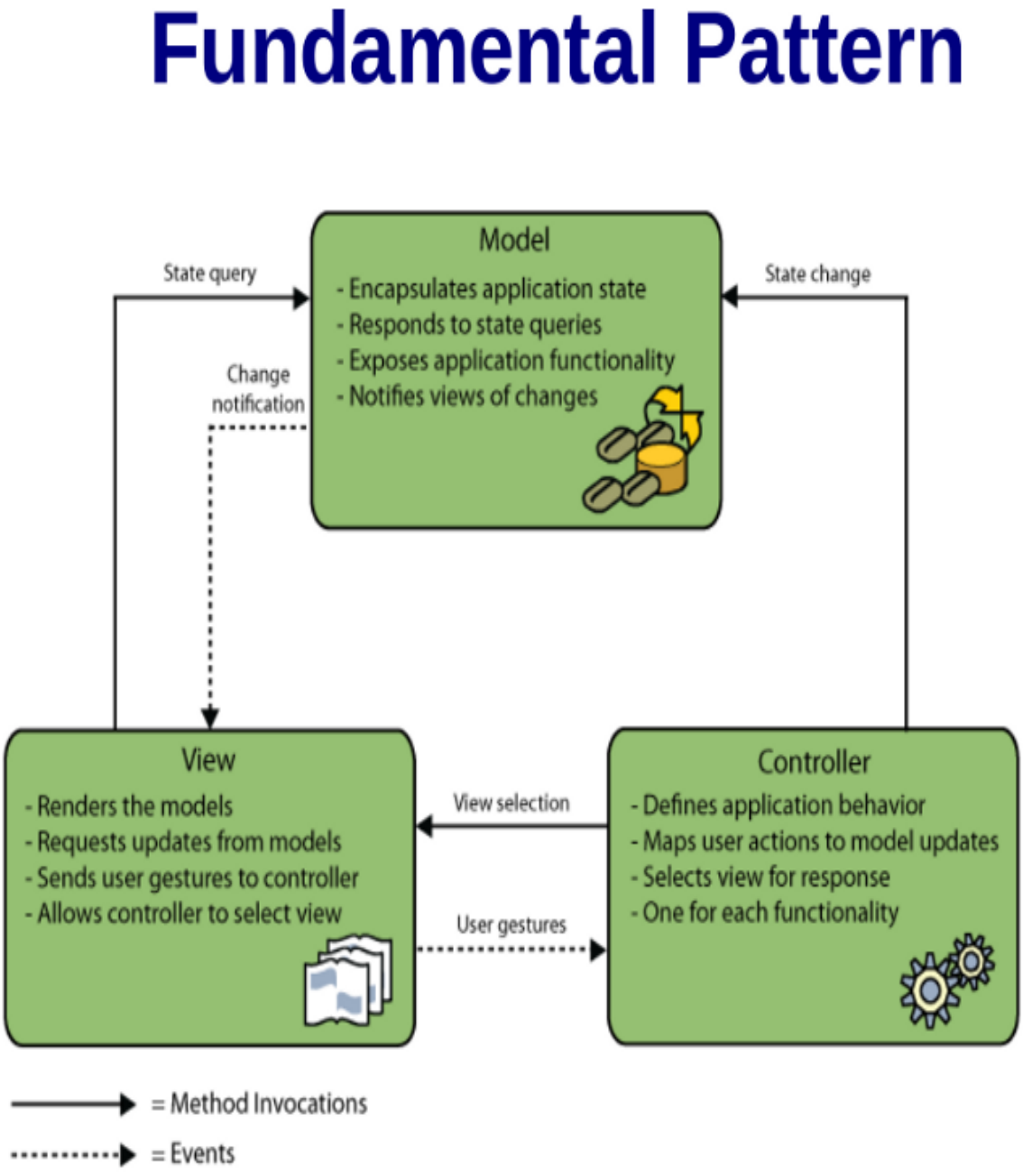
Application: RPC, SOA, REST

Infrastructure: email, messaging, services

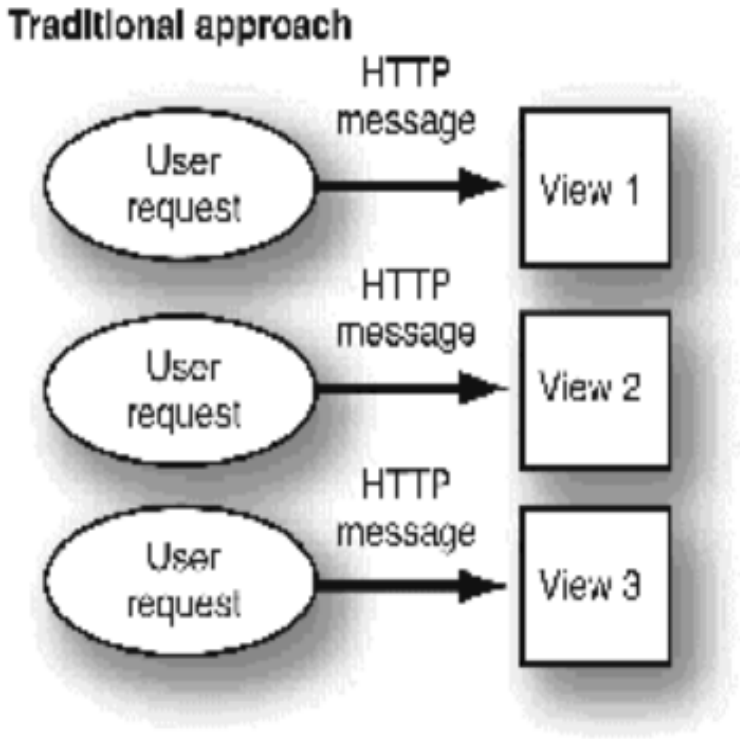
Web Server Basics

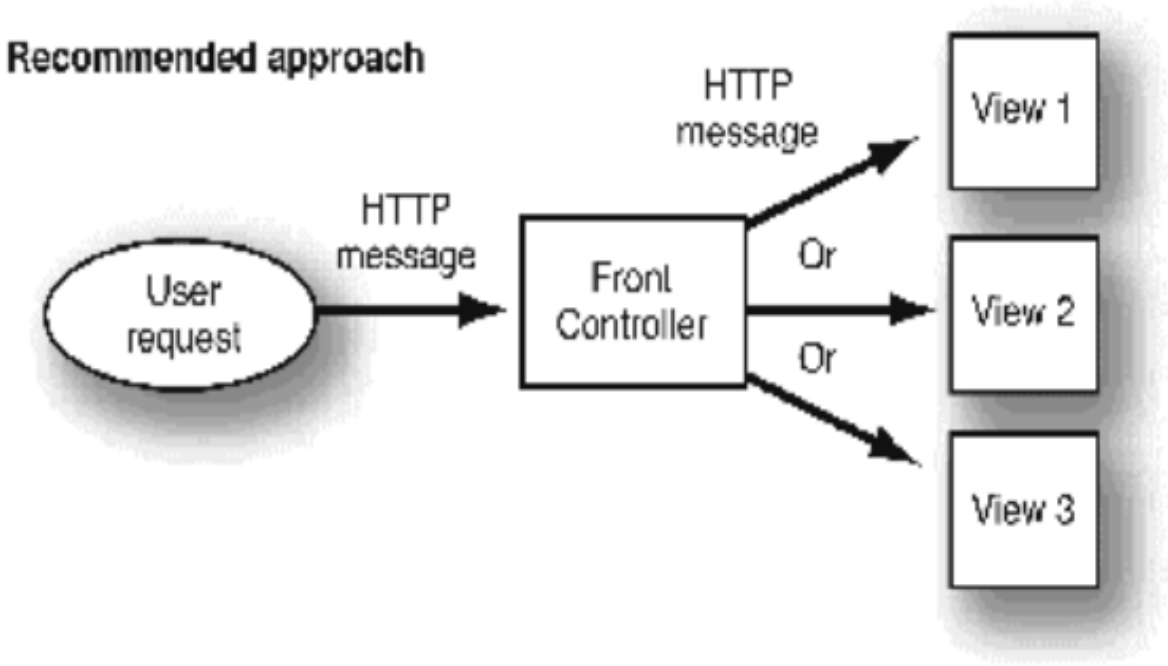


Fundamental Pattern



Approach Evolution



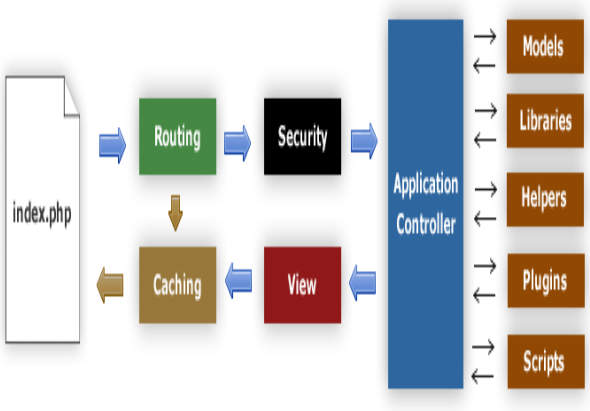


CodeIgniter Architecture

CodeIgniter routing involves request lifecycle "hooks", with security rules, before identifying the right controller.

The controller then uses other components, like helpers, models, libraries, etc, to build what is passed to a view.

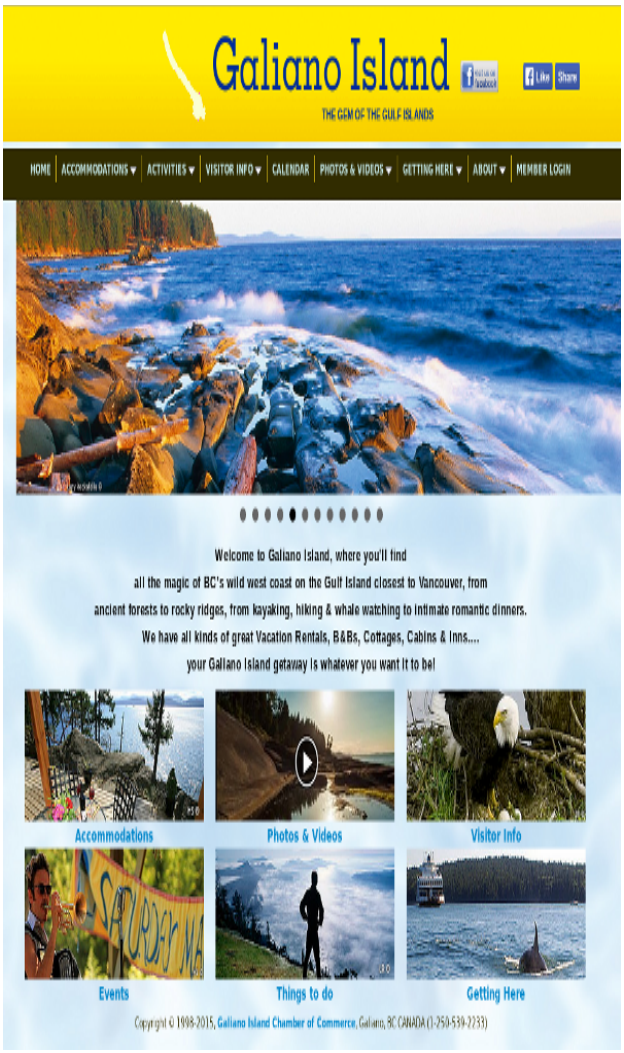
Many such components are built into CI, and the webapp developer would normally build their own by using or extending these.



CodeIgniter Examples

The CodeIgniter website itself is (not surprisingly) implemented using CodeIgniter, and it is open-sourced:)

Another site implemented using CodeIgniter is the Galiano Island tourism site.



Congratulations!

You have completed lesson #intro01: Introduction

If you would take a minute to provide some feedback, we would appreciate it!

The next activity in sequence is: intro02 Golden Rules

You can use your browser's back button to return to the page you were on before starting this activity, or you can jump directly to the course homepage, organizer, or reference page.