Bonus Material

CSCI 4140 – Tutorial 13

Symfony: PHP framework for web projects

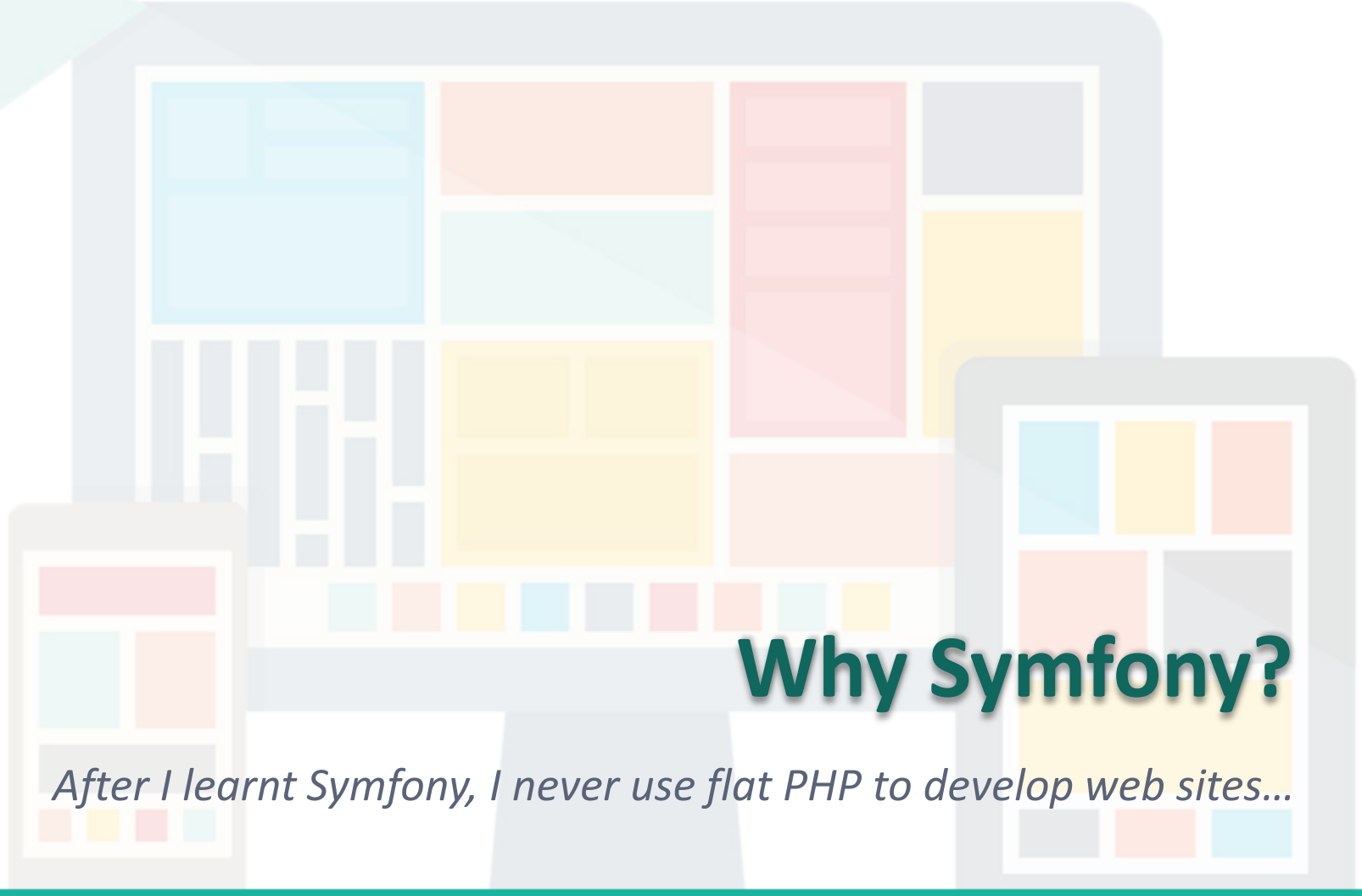
Matt YIU, Man Tung (mtyiu@cse)
SHB 118
Office Hour: Tuesday, 3-5 pm

2015.04.16

Outline

- Why Symfony?
- Demo: Developing a guest book application with Symfony

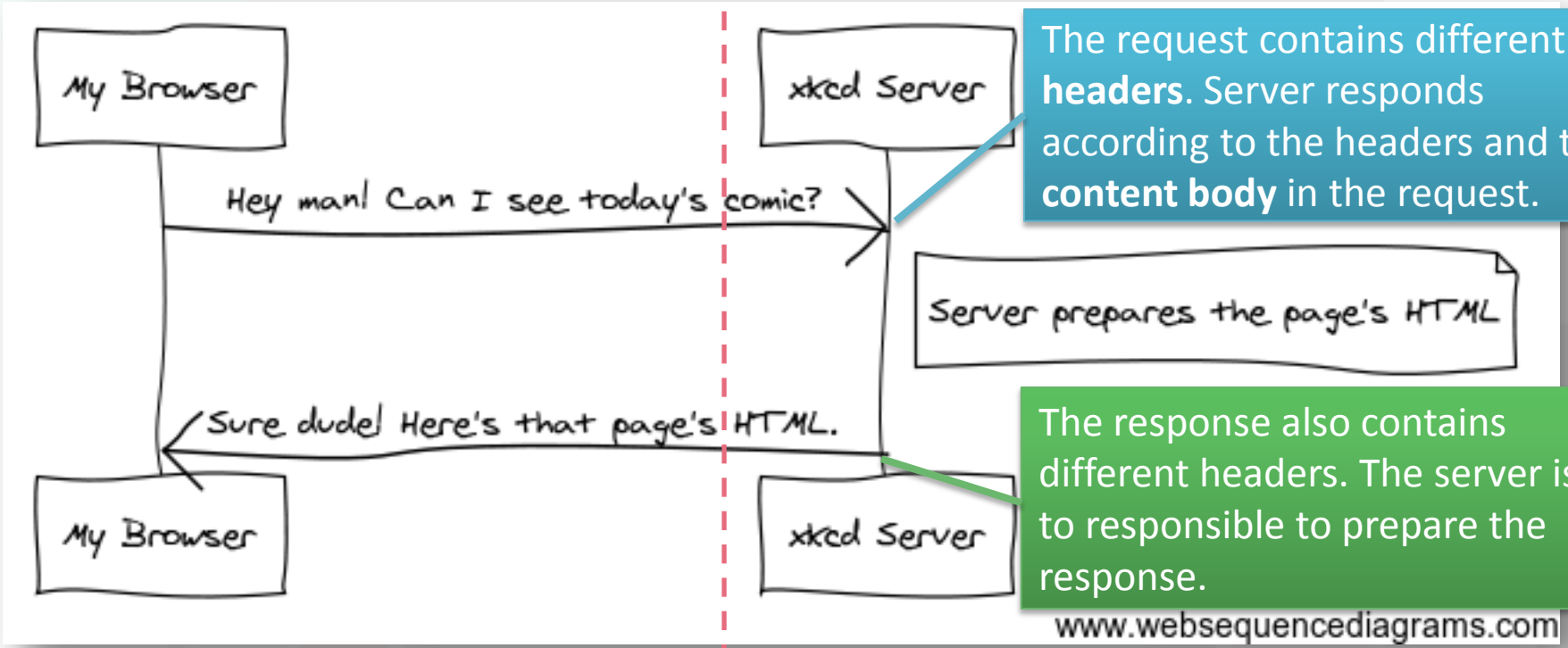
- **Note:** Symfony is not required in this course. In other words, this tutorial covers the bonus material which aims at enhancing your skill sets.



Why Symfony?

After I learnt Symfony, I never use flat PHP to develop web sites…

HTTP is simple!

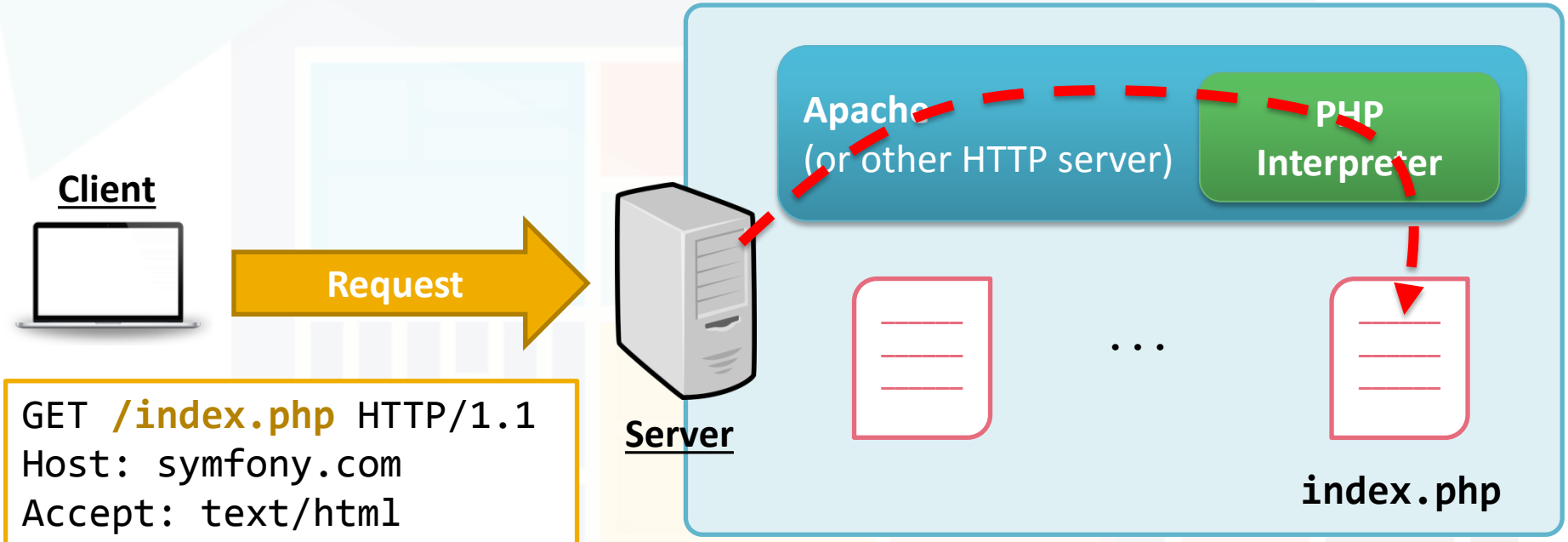


The request contains different **headers**. Server responds according to the headers and the **content body** in the request.

The response also contains different headers. The server is to responsible to prepare the response.

You are experienced in developing the server-side applications in Python and Node.js. In this tutorial, we will use the Symfony framework to do this part.

Requests and Responses in PHP




```
GET /index.php HTTP/1.1
Host: symfony.com
Accept: text/html
User-Agent: Mozilla/5.0
(Macintosh)
```

The request determines which PHP script will be executed.

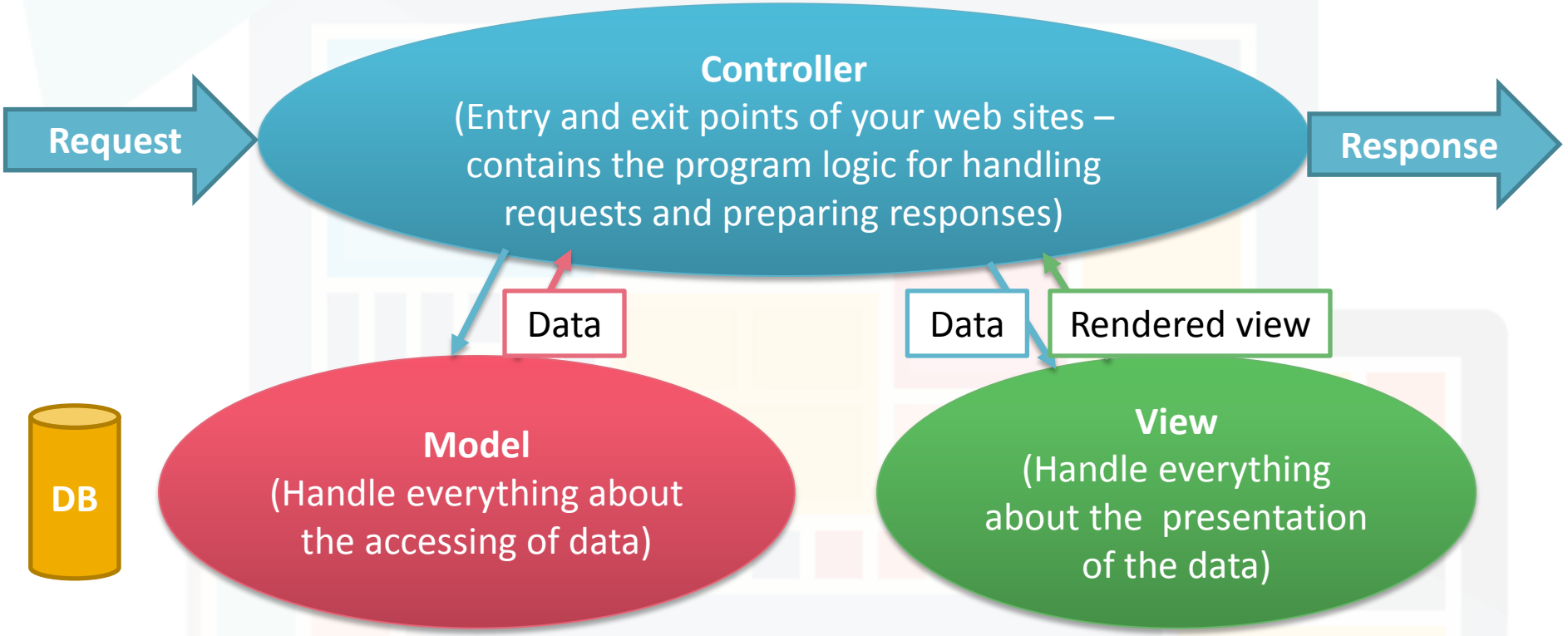
In traditional web development in PHP (similar in Python), the PHP script is responsible to:

- Read the response headers and contents
- Perform operations (e.g., accessing the database)
- Generate the response (views & headers)

Requests and Responses in PHP: Problems

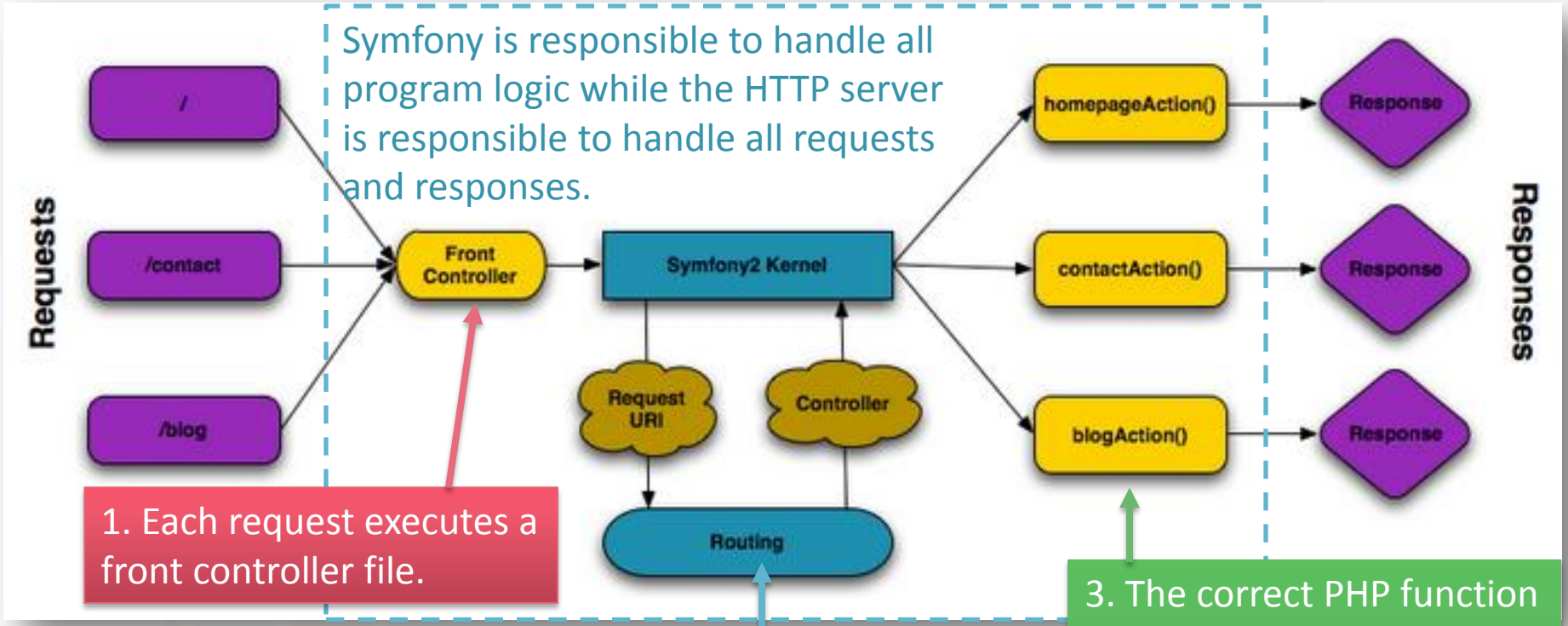
- When you want to modify the file name of a certain PHP script, you need to change ALL the hyperlinks
  - Symfony handles this by using a **front controller** to perform routing
- The PHP script contains all the **program logic** and is responsible to **render the view** → Difficult to maintain!
  - I will show you an example in an popular web site in CUHK…
  - Symfony adopts the **MVC architecture** to separate them *
- Some functionalities are so common that many web applications need them
  - Symfony provides them as **components**
  - **"Don't Repeat Yourself" (DRY)** VS "Write Everything Twice" / "We Enjoy Typing" (WET)

Sidetrack: The MVC architecture

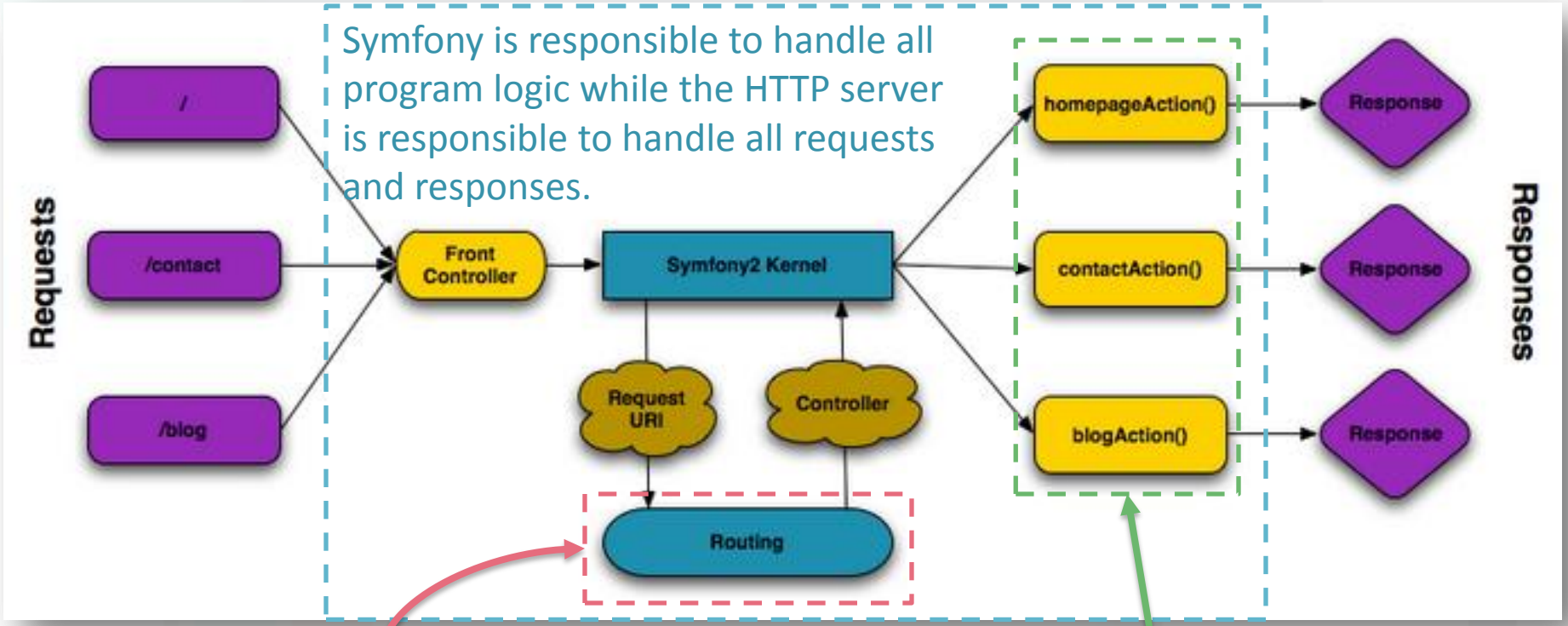


FYI, someone argues that the MVC architecture in most web development framework is not really MVC: http://blog.turn.tw/?p=1539 (in Chinese)

Symfony application flow



Defining routes and implementing controllers



You need to specify the mappings from the request URI to the controller.

The program logic is implemented in the methods inside controllers. It also access the model and view components.

The Symfony components

- Symfony is a collection of independent libraries (called Symfony Components) that can be used inside any PHP project
  - HttpFoundation
  - **Routing**
  - Form
  - **Validation**
  - Templating
  - Security
  - Translation

Routing helps you handle route parameters

Validation helps you validate form input

The Symfony Framework

- A PHP library that
  - Provides a selection of **components** (i.e. the Symfony Components) and **third-party libraries** (e.g. Swift Mailer for sending emails)
  - Provides sensible **configuration** and a **"glue" library** that ties all of these pieces together
- To install other third-party libraries, Symfony uses **Composer** (similar to `npm` in Node.js)

Demo: Developing a guest book application with Symfony

Let's go through the key concepts in Symfony with an example!

Key concepts

- **Environment**
  - Every Symfony application runs within an **environment**
  - **Configuration** + loaded **bundles**
  - Defined **dev** (accessible by the front controller **web/app_dev.php**), **test**, and **prod** (accessible by the front controller **web/app.php**) by default
- **Front controller**
  - Responsible to initialize the **kernel** according to the environment and whether the **debug mode** is turned on
  - Symfony maintains a **cache** (under **app/cache/**)to make your application respond faster
  - Debug mode disables the cache to enable debugging functions

Key concepts

- **Bundle**
  - A bundle is like a **plugin**
  - All the code inside your application will live inside a bundle
    - PHP classes, configuration, templates, …
  - The code of bundles is stored under the directory **`src/`**
  - Register a bundle with the **kernel** (**`app/AppKernel.php`**) to enable it
- **Kernel**
  - The core of Symfony
- **Route**
  - A map from URL path to a controller
  - You already saw this in Express in Assignment 2

Key concepts

- **Template**
  - A text-based file for rendering views
  - Symfony uses **Twig** as the templating engine
- **Object Relational Mapper (ORM)**
  - Sits on top of a powerful **Database Abstraction Layer (DBAL)**
  - Maps database entries to PHP objects to make database accesses easier
  - Symfony supports **Doctrine** and **Propel**

Directory structure

- `app/`
  - Contains application **configuration**
- `src/`
  - Contains all the **project PHP code**
- `vendor/`
  - Stores all **vendor libraries** (by convention)
- `web/`
  - **Web root directory**
  - Contains all **publicly accessible files** (CSS / JavaScript / …)

Please come to the tutorial for demo...

- I will build a guest book application with Symfony
- Only incomplete instructions are available on the Tutorial Resource Page (as I don't have time…)
- Don't worry! This part will be video-taped

Useful bundles

- FOSUserBundle: Add support for a database-backed user system
- KnpPaginatorBundle: SEO-friendly paginator to paginate everything
- DdeboerDataImportBundle: Import data from and store data to a range of formats and media
- FOSRestBundle: Provide various tools to rapidly develop RESTful API & applications
- SonataAdminBundle: Generate admin interface
- *Visit http://knpbundles.com/ to explore more!*

Summary

- Symfony is a powerful web development framework in PHP
- We went through several key components during this tutorial
  - Creating **routes** and the corresponding **controllers**
  - Using the **Twig templating engine**
  - Using the Doctrine ORM to simplify database accesses (**CRUD – Create, Retrieve, Update, Delete**)
  - Making **HTML forms** and performing **form validation**
- There are many interesting topics that we don't have time to cover!
  - Read the Symfony Book and the Symfony Cookbook to learn more

Hope you enjoy the tutorials and assignments ☺

Good luck for your project and final examination!

– End –