

Development of Web Applications

Principles and Practice

Vincent Simonet, 2014-2015

Université Pierre et Marie Curie, Master Informatique, Spécialité STL

7

Web Development Frameworks

Vincent Simonet, 2014-2015

Université Pierre et Marie Curie, Master Informatique, Spécialité STL

Today's agenda

- Programmatic Approaches,
- Template Approaches,
- Hybrid Approaches,
- Model-View-Controller (MVC) Approaches,
- Approaches with a single programming language,
- Stratified Languages.

Web application frameworks

There are hundreds of frameworks on the market. See [Wikipedia](https://en.wikipedia.org/wiki/List_of_web_frameworks) for a (probably non-exhaustive) list.

They differ by various aspects:

- Scope,
- Underlying programming language and technologies,
- **Structure model for the web application,**
- Development model and license.

There are probably some bad choices, but there is no "best" choice.

Typical features of a web application framework

- Web template system,
 - Caching,
 - Access control (authentication, authorization),
 - Database access, mapping and configuration,
 - Scaffolding,
 - URL mapping,
 - Form validation,
 - Ajax,
 - i18n and l10n,
 - Web services (SOAP) and resources (REST).
-

Programmatic Approaches

Programmatic Approaches

The source for a page consists mainly of code written in a scripting or high-level programming language (e.g. Perl, Python or Java).

The body of the code consists of application logic, while the page formatting (HTML) is produced using output statements.

Main limitation: producing HTML using output statements is painful, and cannot be decoupled from programming.

Examples: CGI or Java Servlet (alone).

Template Approaches

Template Approaches

The "inverse" of programmatic approaches: the source for a page consists mainly of formatting structures, with limited embedded constructs containing programming logic.

Appealing for web designers, but limited to implement complex logic.

Examples: SSI, Apache Velocity.

SSI Server Side Includes

A simple interpreted server-side scripting language. Supported in `.shtml` files by Apache and IIS.

```
<!--#include virtual="../quote.txt" -->
<!--#exec cmd="ls -l" -->
```

Very simple, but not very powerful.

Apache Velocity

Java-based template engine that provides a template language to reference objects defined in Java code.

Used for HTML pages, but also for source code generation, automatic emails, XML transformation.

Apache Velocity: Example

```
## Velocity Hello World
<html>
  <body>
    #set( $foo = "Velocity" )
    ## followed by
    Hello $foo World!
  </body>
</html>
```

Hybrid Approaches

Hybrid Approaches

In hybrid approaches, scripting elements and template structures are combined in the same source file, with the same level of importance.

Main limitation: designers and developers must work on the same source objects.

Examples: PHP, ASP.NET, JSP (alone).

PHP PHP: Hypertext Preprocessor



Server-side scripting language. PHP code can be directly embedded into HTML pages. It is interpreted by the server at serving time.

Free software. More than 244M websites on 2.1M webserver.

PHP is part of the LAMP stack. Many web application frameworks have also built on top of PHP (e.g. Symfony).

PHP: Example

```
<html>
  <body>
    <h1>
      <?php if ($title != "") {
        print $title;
      } else {
        ?>Default Title<?php } ?>
    </h1>
    ...
  </body>
</html>
```

ASP.NET

The web development framework from Microsoft, part of the .NET Framework.

Similarly to PHP, code (in any .NET supported language, typically VB or C#) can be inlined into HTML pages.

Contrary to PHP, HTML templates are compiled.

ASP.NET Extensions

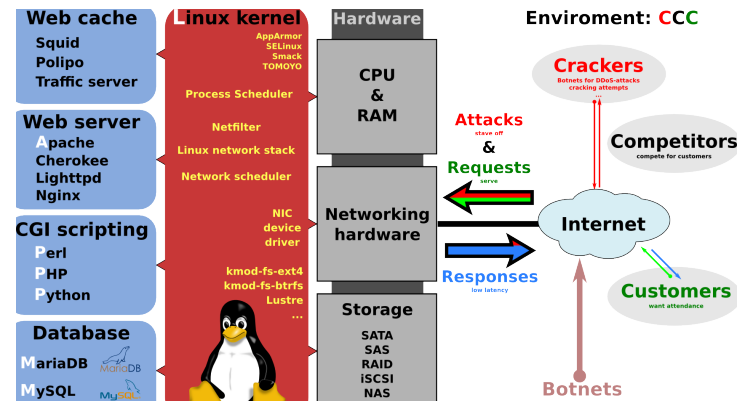
- **ASP.NET AJAX:** Server framework + JavaScript client library for AJAX communication.
- **ASP.NET MVC Framework:** Model-View-Controller framework on top of ASP.NET
- **ASP.NET Web API:** HTTP API for exposing web services.
- **ASP.NET SignalR:** real-time client/server communication framework.

ASP.NET: Example

```
<%@ Page Language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 //EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
    protected void Page_Load(object sender, EventArgs e)
    {
        // Assign the datetime to label control
        lbl1.Text = DateTime.Now.ToString();
    }
</script>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Sample page</title>
</head>
<body>
    <form id="form1" runat="server">
        ...
```

LAMP



Model-View-Controller Approaches

Model-View-Controller

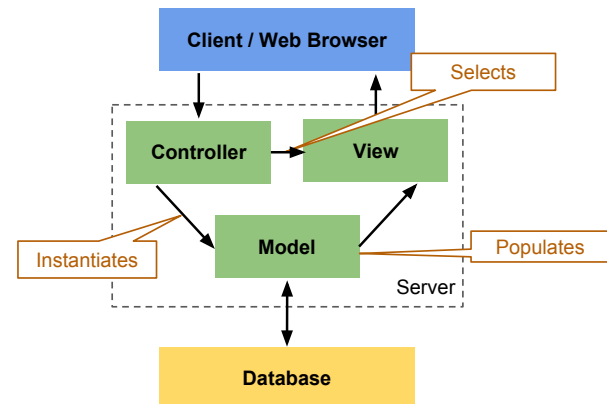
A software architecture, used by many modern web development frameworks (but not specific to web).

- **Model:** application data, business rules, logic, and functions.
- **View:** any output representation of data.
- **Controller:** a mediator for input, converting it to commands for the model or view.

Model-View-Controller in web applications

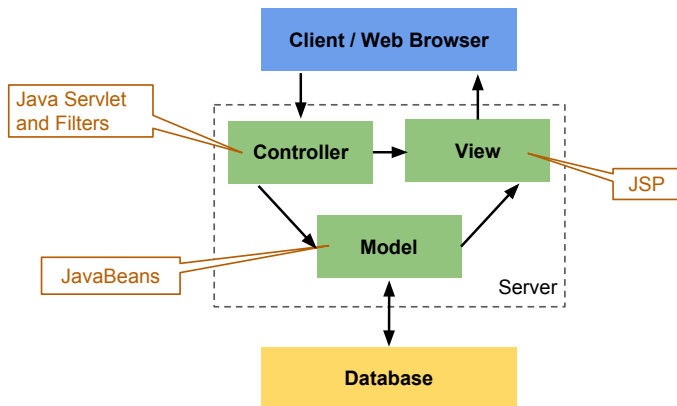
- HTTP requests are directed to the **controller**,
- The controller access required data and instantiates the **model**, typically a POD datastructure,
- The controller selects, populates and passes control to the appropriate **view**, typically a template page with some programming constructs accessing data in the model,
- The view page is rendered and sent as an HTTP response.

Model-View-Controller in web applications

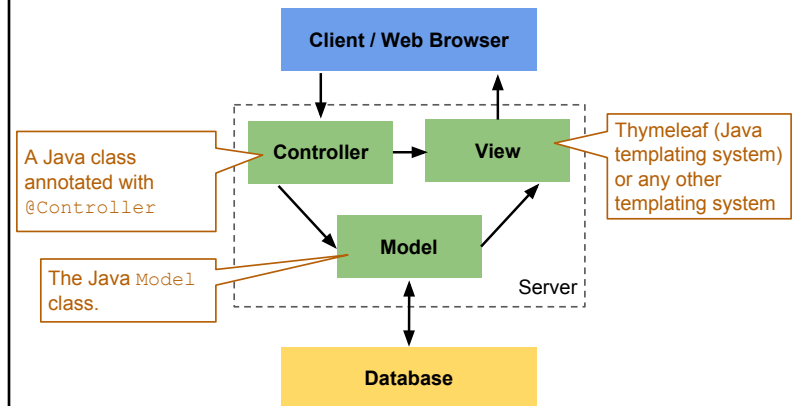


Model 2 / Apache Struts

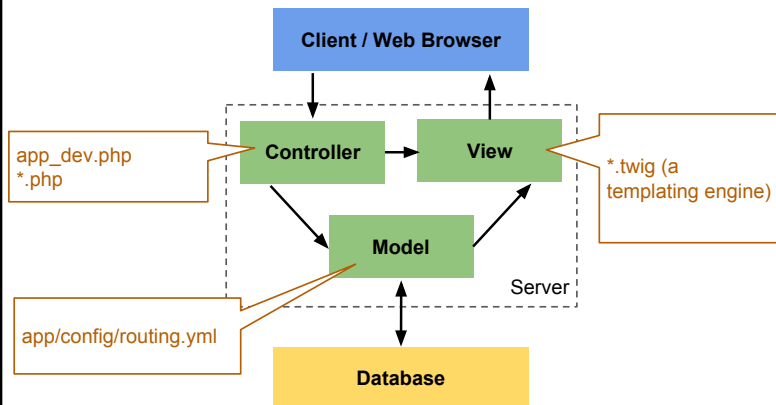
Struts



Spring

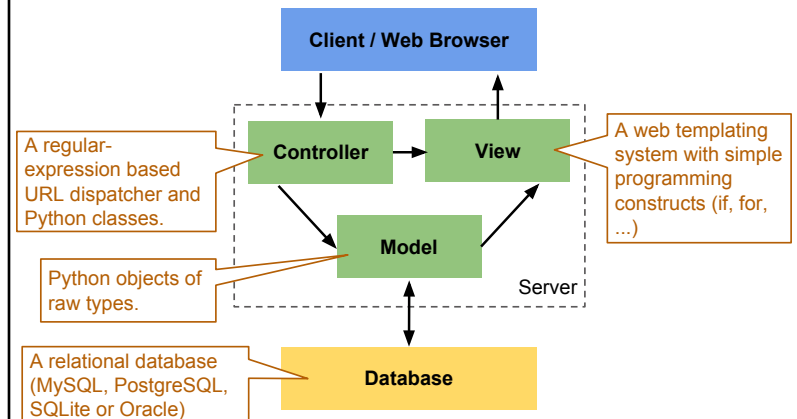


Symfony

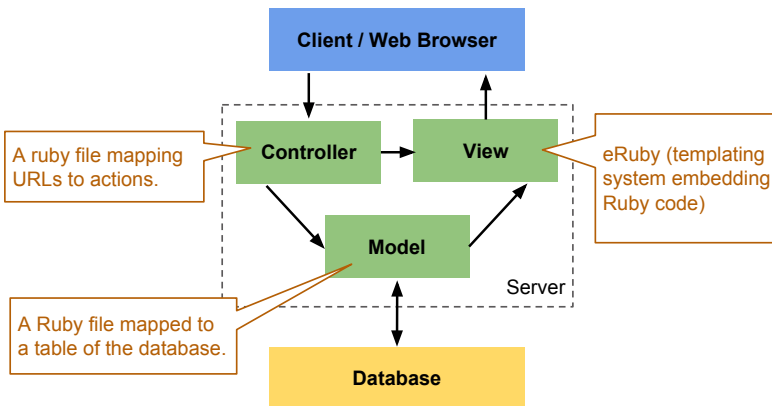


Django

django



Ruby on Rails



Push- vs pull-based MVC

The previous MVC frameworks are said to be **push-based**: a user action triggers the required processing, and data is then pushed to the view layer to render results.

Pull-based is an alternative approach, where the processing starts with the view layer which can then pull results from multiple controllers as needed. Examples: JavaServer Faces, Lift, Wicket.

Angular JS



A JavaScript framework for developing single-page applications using a MVC approach on the client side.

The application is defined in an HTML file with custom tag attributes, which are read and processed by the framework. The framework binds input and output parts of the page to a model represented by standard JavaScript variables.

Approaches with a single programming language

What's up?

For historical reasons, different (programming) languages are generally used on server and client sides.

This makes the programming of a web application more complicated, prevents re-using code, etc.

GWT Google Web Toolkit



Develop AJAX applications in Java.

Main components:

- **GWT Java-to-JavaScript Compiler:** Translates the Java programming language to the JavaScript programming language.
- **GWT Development Mode:** Allows the developers to run and execute GWT applications without compiling into JavaScript.
- **JRE emulation library:** JavaScript implementations of JRE standard library.
- **GWT Web UI class library:** A set of custom interfaces and classes for creating widgets.

Used by Google (Blogger, AdWords, Groups, etc.) and other companies.

Node.js

As we have JavaScript on the client side, let's have JavaScript on the server side!

Node.js is an efficient JavaScript interpreter (V8) that can run within an HTTP server, or that can run an HTTP server.

There are tens of [web application frameworks](#) based on Node.js. The most popular are Express and Geddy (MVC).

Stratified Languages

What's up?

Use the same programming language for client and server, in a **single source code**.

In most cases, the programmer has to explicitly specify what has to be executed on the server and what has to be executed. But one could imagine to delegate this task to the compiler (though this is not easy, *e.g.* for security or performance reasons).

Hop



A Lisp-like stratified programming language, developed by Manuel Serrano (INRIA Sophia-Antipolis) using Bigloo Scheme.

Hop: Example

```
(define-service (server-date)
  (current-date))

(<HTML>
  (<BUTTON>
    :onclick ~(with-hop ($server-date)
                  (lambda (h) (alert h)))
    "Server time"))
```

HTML tags are functions

~ means the client should process

\$ means that server-date should be evaluated on server before sending the HTML to the client

HTML attribute

RPC call to the server

Ocsigen



OCaml-based web development framework, consisting of:

- **Eliom**: a framework for programming web sites and client/server web applications, the core of the system,
- **js_of_ocaml**: an OCaml,
- **Ocsigen Server**: an HTTP server, using
- **lwt**: a cooperative threading library.

Ocsigen: Example

```
let mywidget s1 s2 =
  let button = div ~a:[a_class ["button"]] [pdata s1] in
  let content = div ~a:[a_class ["content"]] [pdata s2]
  in
  {unit{ async (clicks %button
                (fun _ _ -> switch_visibility %content))
  }};
  div ~a:[a_class ["mywidget"]] [button; content]

let _ =
  register_service ~path:[] ~get_params:Eliom_parameter.
  unit
    (fun () () ->
     return (Eliom_tools.D.html ~title:"ex"
                                ~css:[["css"; "ex.css"]]
                                (body [h2 [pdata "Welcome to Ocsigen!"];
                                       mywidget "Click me" "Hello!"])))
```

Opa



- A single Opa program is compiled to JavaScript for the client and the server (Node.js).
- Functional language with static type system and type inference.
- Message passing similar to Erlang

Some natural evolution...

