Western Norway University of Applied Sciences

Web Development: Frameworks

Created by Lasse Jenssen

(Based on material from Atle Geitung, 2021)

Home

Agenda: Using Web Frameworks

- What is a framework?
- Pros and cons: Why use frameworks?
- Types of Web Framworks
- Action Based Web Frameworks
- Component Based Web Framework
- Introduction to Spring Web MVC (later)
- Introduction to Thymeleaf (later)

Introduction to Web Framworks

What is a Framwork?

Definition from wiki: http://en.wikipedia.org/wiki/Framework

"A framework is a generic term commonly referring to an essential ***supporting structure*** *which* ***other things are built on top of****."*

Introduction to Web Framworks

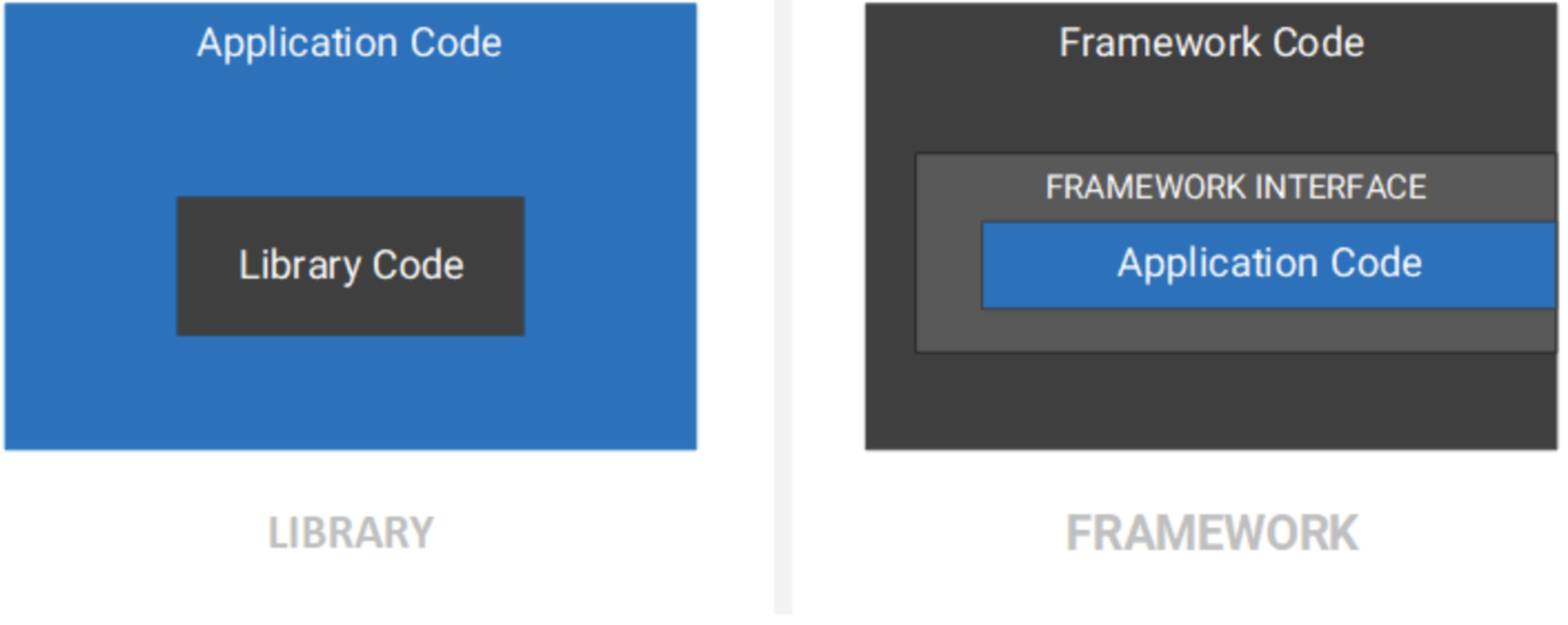
What is a Web Framwork?

Definition from wiki: https://en.wikipedia.org/wiki/Web_framework

*"A software framework that is designed to **support the development of web applications** including web services, web resources, and web APIs.*
*Web frameworks provide a **standard way to build and deploy** web applications on the World Wide Web, and aim to **automate the overhead associated with common activities** performed in web development."*

Introduction to Web Framworks

Libary vs Framework



Source: https://dzone.com/articles/dependency-injection-in-spring

Introduction to Web Framworks

Pros:

- Supposed to simply the development.
- Faster and more robust code (but faster is not always better).
- Common architecture that project/ team/ department follows.
- Less own code to maintain.
- Comunitity: Get help or assistance from other users.
- Offers:
  - Easy and flexible way to configure (XML, Annotations, property files etc)
  - Tab library (for instance validating input in Ajax)
  - Class libraries
  - And much more ...

Introduction to Web Framworks

Cons:

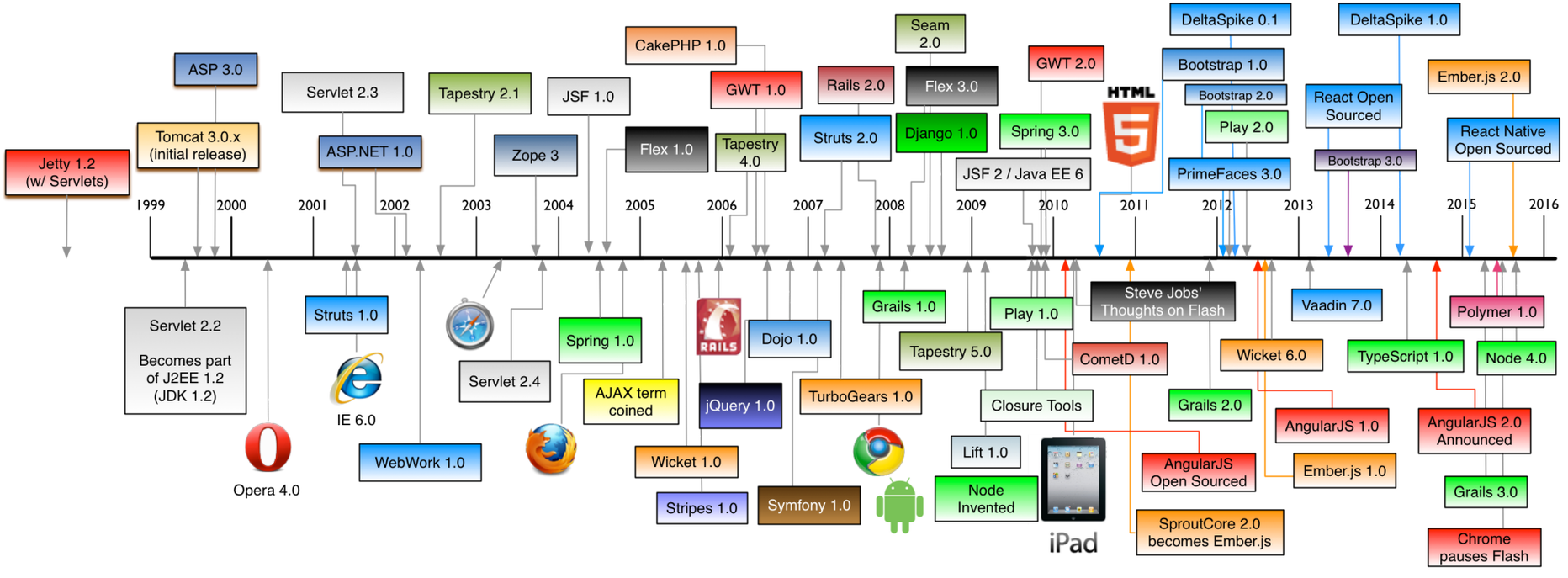
- High startup cost: a lot to learn.
- Many frameworks to choose from.
- Framework implementation might not be the best approach for ALL your problems.
- Dependency/ "lock-in".

Types of Web Frameworks

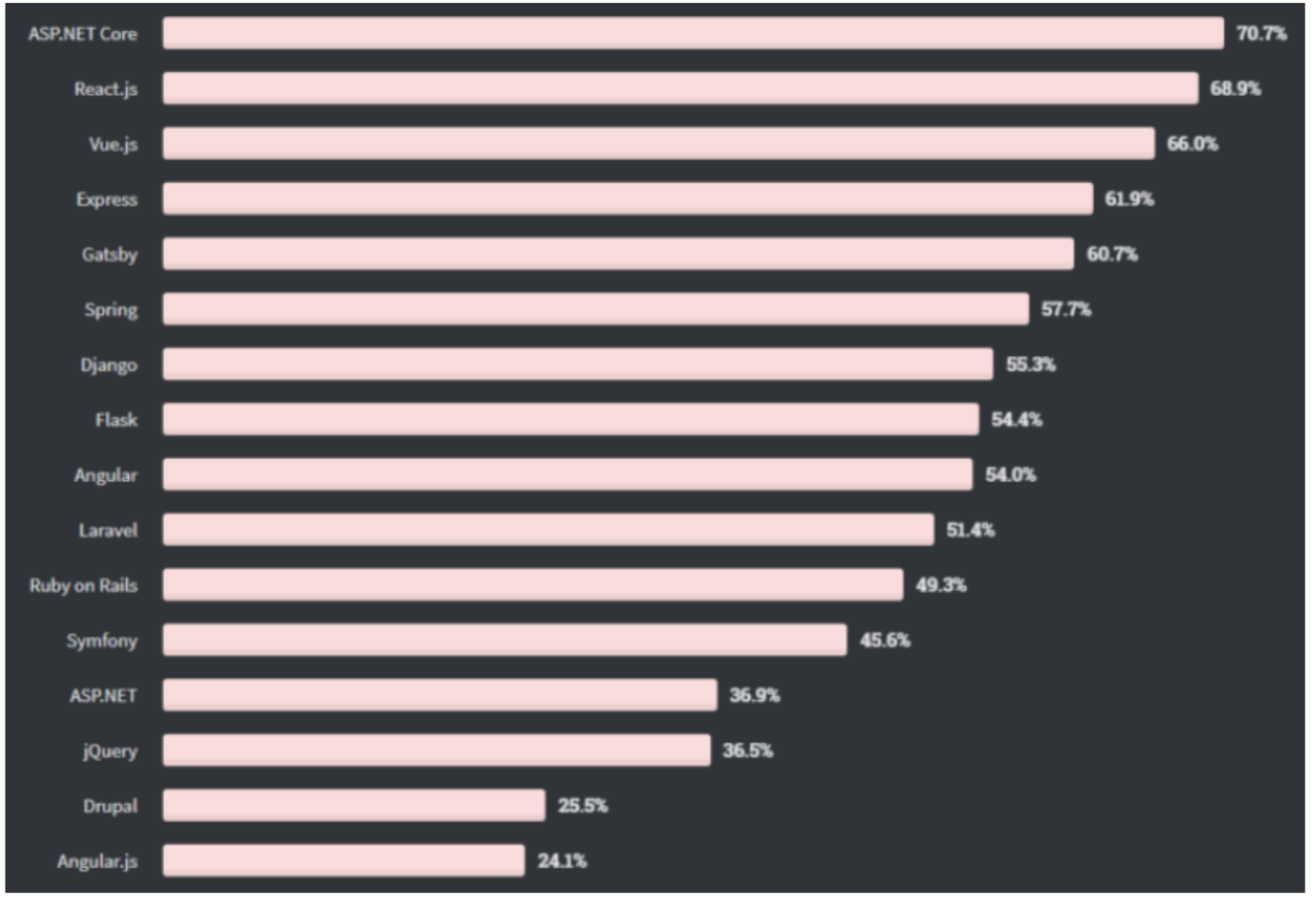
- Types:
  - **Client-side Web Frameworks**: For instance: Angular, Vue, React (you have looked at React earlier).
  - **Server-side Web Frameworks**: Todays agenda.

Types of Web Frameworks

- A common classification of web frameworks is based on the "programming model" being offered the developer.
- Web applications have a core with a programming model based on the **request-response** over a stateless protocol, HTTP (Servlet).
- Types:
  - **Action-based**: Frameworks that exposes the request-response model, linking "actions" up to URLs (requests).
  - **Component-based**: Frameworks that hides the request-response model behind UI components (+events).
  - Also more specialized frameworks, for example RIA (Client-Side: Angular, React, Vue).
- Note! Not always a clear distinction.


Jetty 1.2
(w/ Servlets)
Tomcat 3.0.x
(initial release)
ASP 3.0
Servlet 2.3
ASP.NET 1.0
Tapestry 2.1
Zope 3
JSF 1.0
Flex 1.0
CakePHP 1.0
GWT 1.0
Tapestry
4.0
Rails 2.0
Struts 2.0
Flex 3.0
Django 1.0
Seam
2.0
GWT 2.0
Spring 3.0
JSF 2 / Java EE 6
HTML
DeltaSpike 0.1
Bootstrap 1.0
Bootstrap 2.0
Play 2.0
PrimeFaces 3.0
DeltaSpike 1.0
React Open
Sourced
Bootstrap 3.0
Ember.js 2.0
React Native
Open Sourced
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
Servlet 2.2
Becomes part
of J2EE 1.2
(JDK 1.2)
Opera 4.0
Struts 1.0
IE 6.0
WebWork 1.0
Servlet 2.4
Spring 1.0
AJAX term
coined
Wicket 1.0
Stripes 1.0
RAILS
Dojo 1.0
jQuery 1.0
Symfony 1.0
Grails 1.0
Tapestry 5.0
TurboGears 1.0
Node
Invented
Play 1.0
Closure Tools
Lift 1.0
iPad
Steve Jobs'
Thoughts on Flash
CometD 1.0
AngularJS
Open Sourced
SproutCore 2.0
becomes Ember.js
Vaadin 7.0
Wicket 6.0
Grails 2.0
AngularJS 1.0
Ember.js 1.0
TypeScript 1.0
Polymer 1.0
Node 4.0
AngularJS 2.0
Announced
Grails 3.0
Chrome
pauses Flash

Use of frameworks 2022



https://www.monocubed.com/blog/most-popular-web-frameworks/

Action based Web Frameworks (MVC)

Examples of action-based framworks:

- Struts2 (Java)
- Spring Web MVC (Java/ Kotlin/ Groovy)
- PlayFramework (Java / Scala)
- Rails (Ruby)
- Grails (Groovy)

Component Based Web Frameworks

Examples of component-based frameworks:

- Vaadin (Java) : We'll have a quick look at this.
- GWT (Java) : GWT is a development toolkit for building and optimizing complex browser-based applications.
- ASP.NET WebForms (MVVM): a part of the ASP.NET web application framework

Component Based Web Frameworks

Also some Component-based Desktop Frameworks:

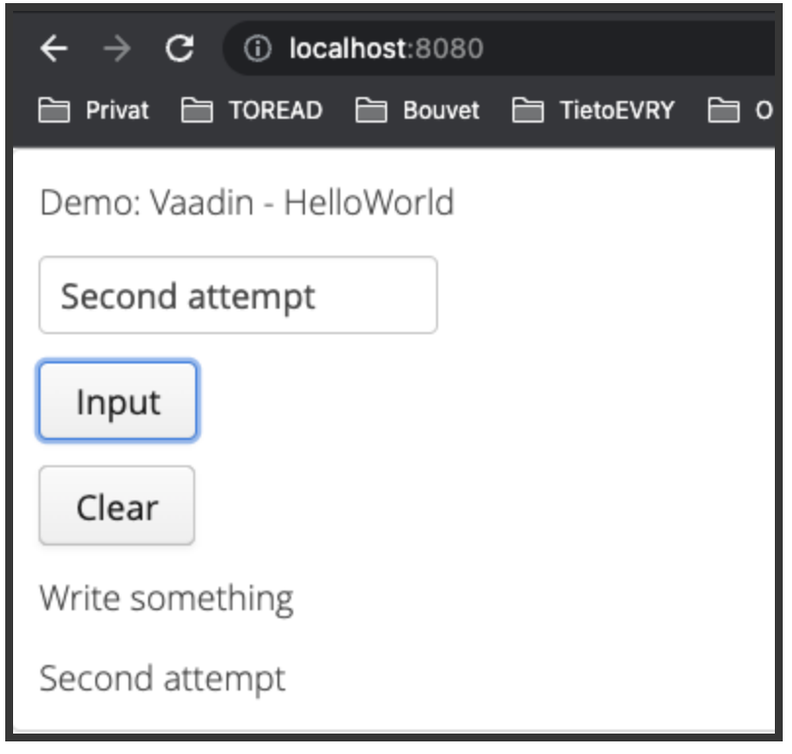
- **Java Swing**: a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) - an API for providing a graphical user interface (GUI) for Java programs.
- **JavaFX**: (https://openjfx.io/) - an open source, next generation client application platform for desktop, mobile and embedded systems built on Java.
- **Java SWT** (https://www.eclipse.org/swt/): an open source widget toolkit for Java designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented.
- **Qt** - (C++, Python, etc)

Demo: HelloWorldUI (Vaadin)

- Intro application with a few simple components (Label, TextField, Button).
- We are NOT going to code with Vaadin.
  (Code in this section: Only to show the concept).

Code: demo-component-based-vaadin.zip (see course overview).

Demo: HelloWorldUI (Vaadin)



Demo: HelloWorldUI (Vaadin)

src/main/java/no/hvl/dat152/helloworld/HelloWorldUI.java

```
1  @Theme("mytheme")
2  public class HelloWorldUI extends UI {
3
4     private static final long serialVersionUID = 1L;
5
6     @Override
7     protected void init(VaadinRequest vaadinRequest) {
8        Panel contentPane = new Panel();
9        VerticalLayout layout = new VerticalLayout();
10
11       Label lblHeading = new Label();
12       helloworldLabel.setValue("Demo: Vaadin - HelloWorld");
13       layout.addComponent(lblHeading);
14       ...
15       contentPane.setContent(layout);
16       setContent(contentPane);
17    }
18
19    @WebServlet(urlPatterns = "/*", name = "HelloWorldUIServlet", asyncSupporte
20    @VaadinServletConfiguration(ui = HelloWorldUI.class, productionMode = false
21    public static class HelloWorldUIServlet extends VaadinServlet {
22       private static final long serialVersionUID = 1L;
23    }
24 }
```

Demo: HelloWorldUI (Vaadin)

src/main/java/no/hvl/dat152/helloworld/HelloWorldUI.java

```
1  @Override
2  protected void init(VaadinRequest vaadinRequest) {
3        Panel contentPane = new Panel();
4        VerticalLayout layout = new VerticalLayout();
5
6        Label lblHeading = new Label();
7        lblHeading.setValue("Demo: Vaadin - HelloWorld");
8
9        TextField textField = new TextField();
10       textField.setValue("Write something");
11
12       Button okButton = new Button("Input");
13       ...
14
15       Button clearButton = new Button("Clear");
16       ...
17
18       layout.addComponent(lblHeading);
19       layout.addComponent(textField);
20       layout.addComponent(okButton);
21       layout.addComponent(clearButton);
22
23       contentPane.setContent(layout);
24       setContent(contentPane);
```

Demo: HelloWorldUI (Vaadin)

src/main/java/no/hvl/dat152/helloworld/HelloWorldUI.java

```
@Override
protected void init(VaadinRequest vaadinRequest) {
      ...
      Button okButton = new Button("Input");
      okButton.addClickListener(event -> {
         Label newLabel = new Label();
         newLabel.setValue(textField.getValue());
         layout.addComponent(newLabel);
      } );

      Button clearButton = new Button("Clear");
      clearButton.addClickListener(event -> {
         layout.removeAllComponents();
         layout.addComponent(lblHeading);
         layout.addComponent(textField);
         layout.addComponent(okButton);
         layout.addComponent(clearButton);
      } );

      layout.addComponent(lblHeading);
      ...
      contentPane.setContent(layout);
      setContent(contentPane);
}
```

Demo: HelloWorldUI (Vaadin)

Let's look at the code in Eclipse, and run.

Code: demo-component-based-vaadin.zip (see course overview).

Summary: Web Development: Frameworks

Where are we now?

- Frameworks are there to assist in our daily programming life.
- A lot of frameworks to choose from.
- Most often the project or company choose the frameworks to be used.

Next

Web Services: The Spring Framework

Home