Module d'expertise de 2e année

Développer des applications full-web : Devenir développeur full-stack !

< 1. Introduction />

– ESTIA –
Guillaume Rivière
Dernière révision : mars 2019

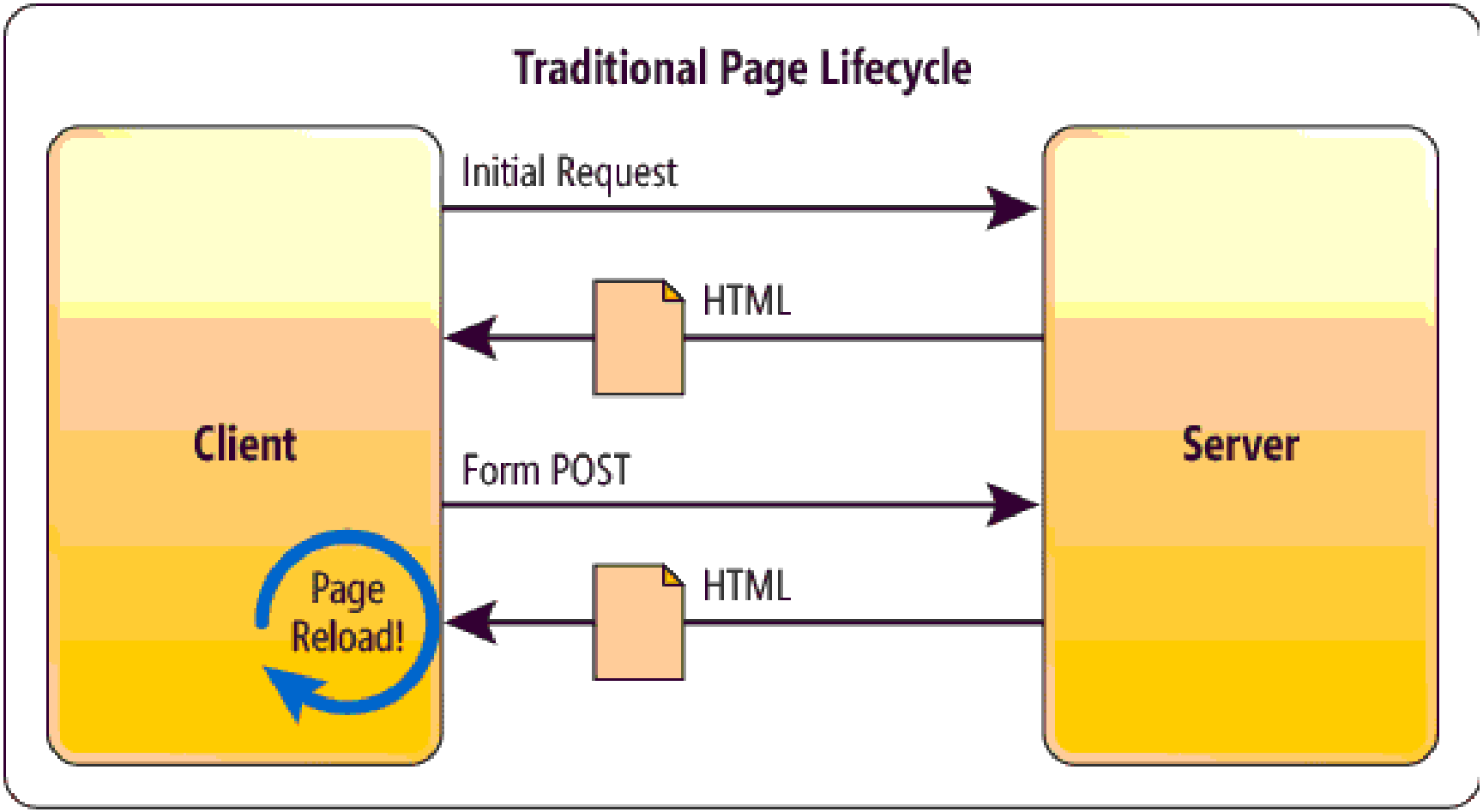
Contexte

- Technologies des GAFAM
  - Permet de déployer des applications et des services à l'échelle mondiale
- Application riches
- Progressive webapps
- Single-Page Application (SPA)

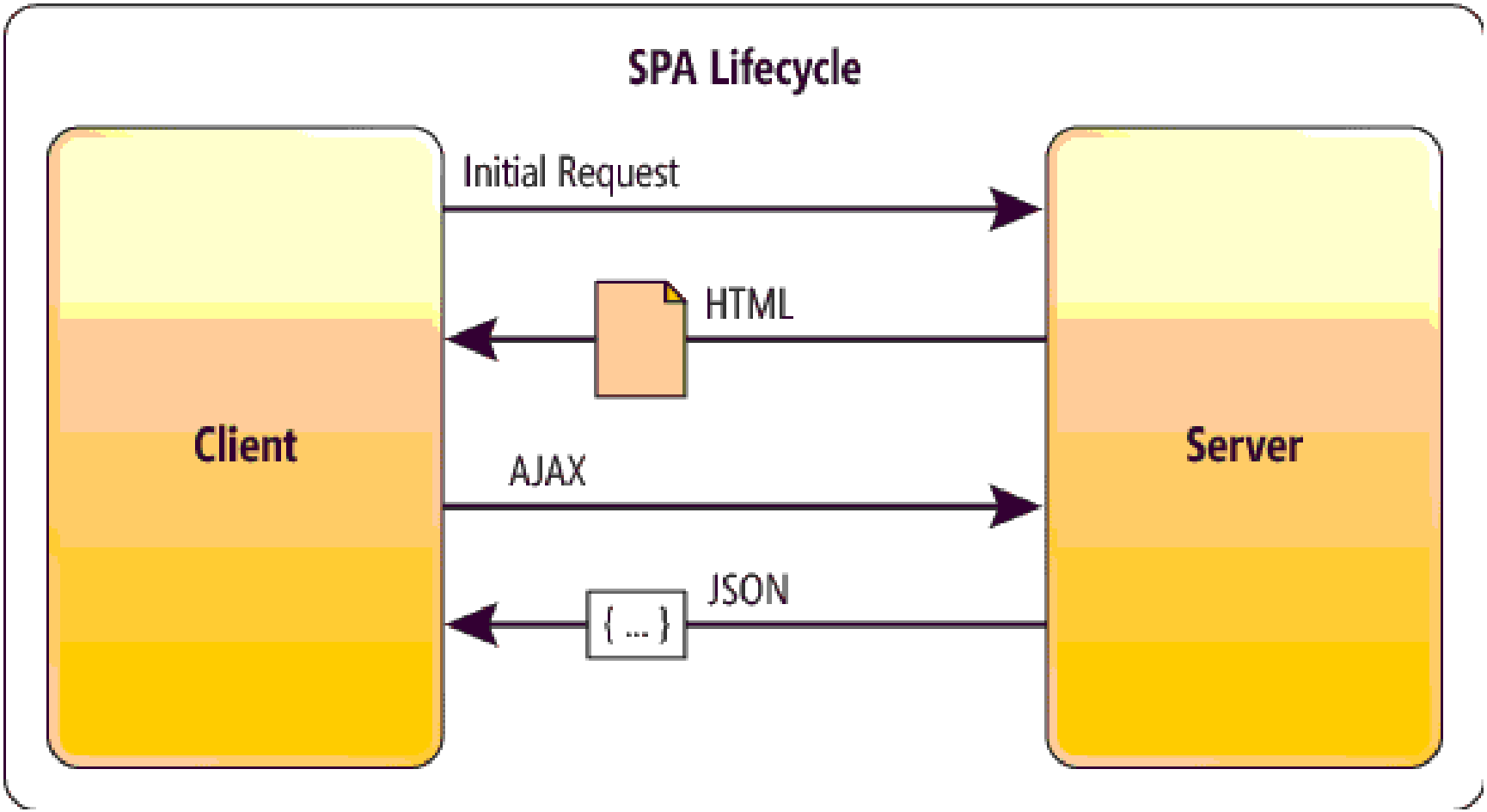
Références

- Youtube
- Dooble
- GMail
- Amazon
- Facebook
- MS O365
- Twitter
- Pinterest
- Onshape.com
- Odoo (OpenERP)

Transactions pages web classiques



Transactions application web : SPA



Histoire

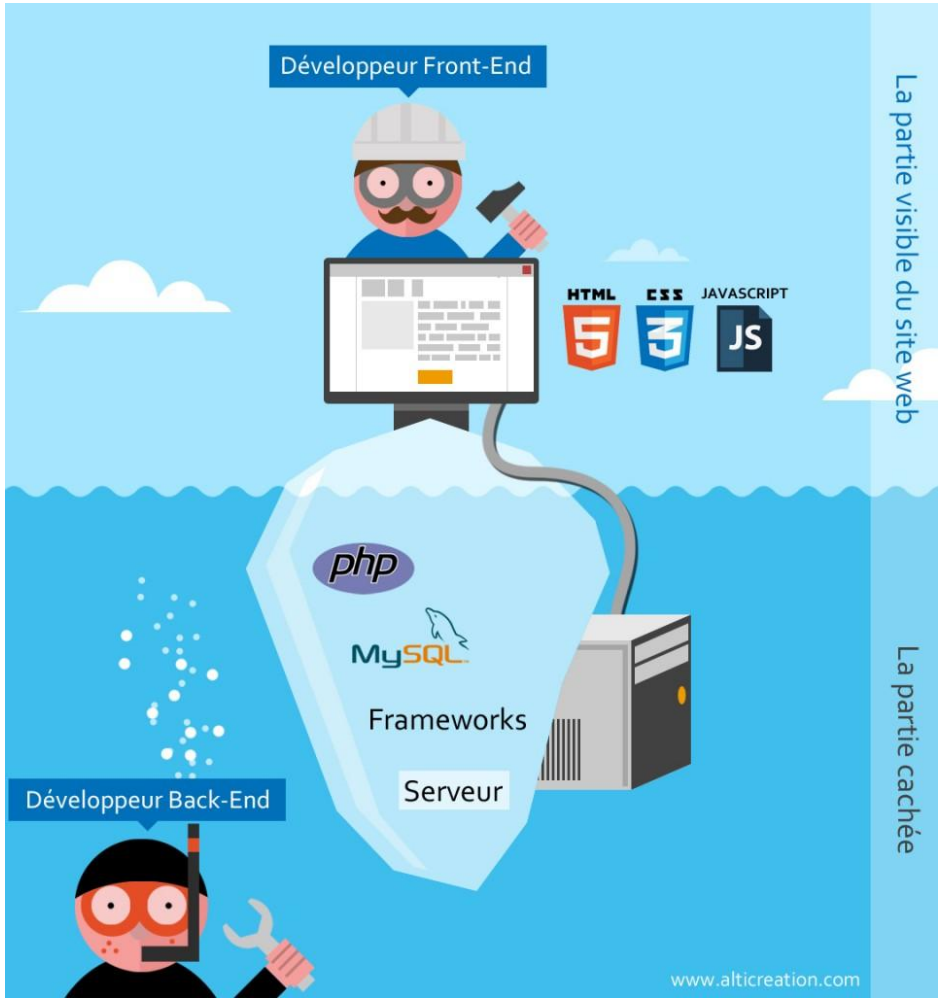
- HTML
- HTML + CSS
- XHTML4
- HTML5 + CSS3
- Ajax
- Jquery
- Bootstrap
- AngularJS / Angular / React / Backbone.js

Application full-web

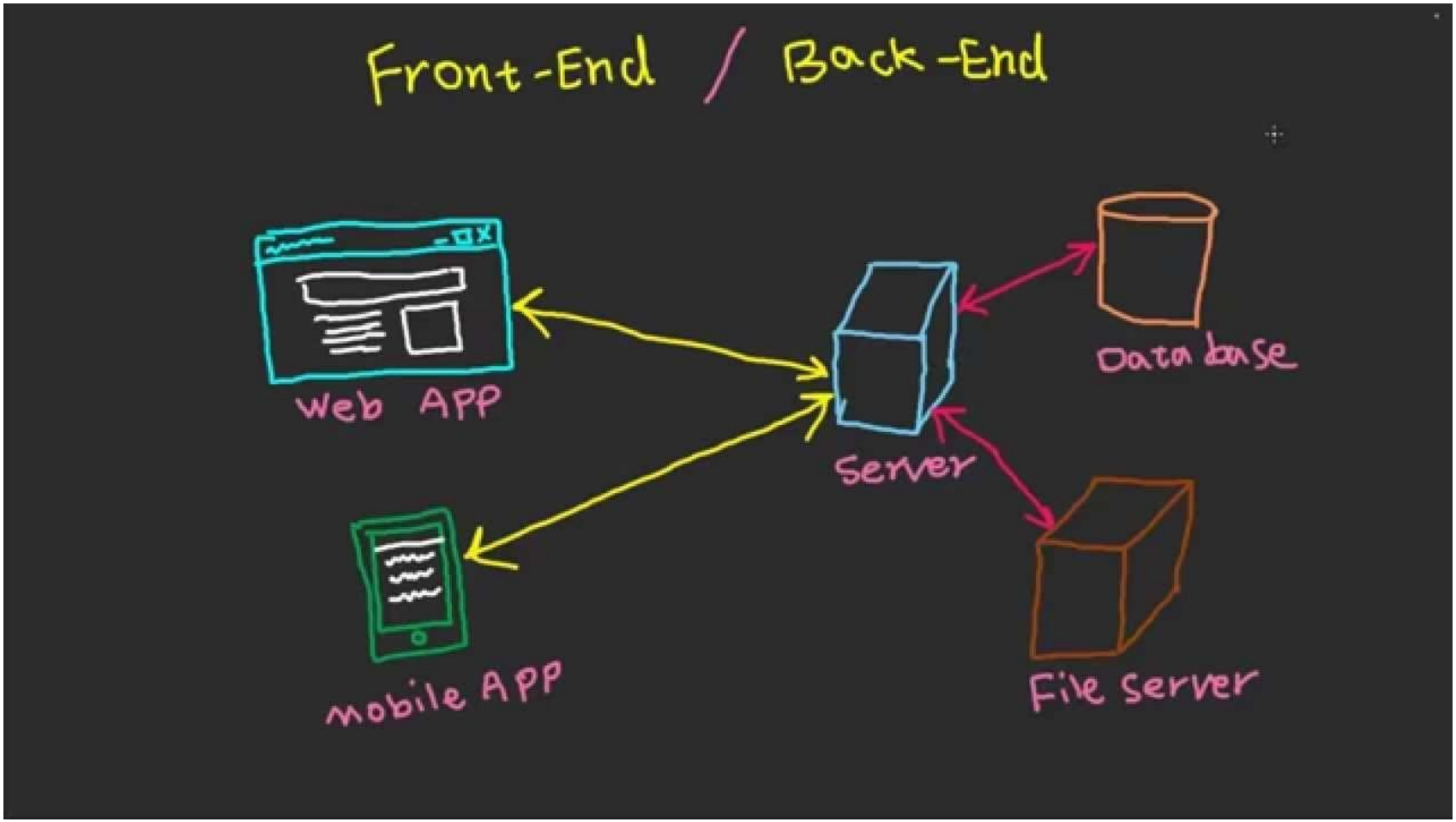
- « Stack »
  - Front-end
  - API / webservice
  - Back-end
  - Base de données

Développeur front-end
||
Développeur back-end

Développeur full-stack



Webservice d'API : applis web / mobile



Cloud computing

- SaaS

- Google Cloud Platform
- AWS
- MS Azure

Front-end

- Angular (TypeScript)
- React
- Vue

Office québécois de la langue française :
Application frontale

Back-end

- Back-end
  - PHP
    - Symfony
    - Laravel
  - Javascript / Typescript
    - NodeJS
  - Python
    - Djando
  - Ruby
    - Rails
  - R
  - C++
    - Crow
    - Silicon
    - Cppcms
    - Tntnet
    - ctml

Office québécois de la langue française :
Application dorsale

Base de données

- Base de données
  - Relationnelle
    - MySQL
    - PostgreSQL
  - Not Only SQL (NOSQL)
    - PostgreSQL
    - MongoDB

Full-Stack

- API / Webservice
  - REST
  - Format
    - JSON
    - XML
    - XML-RPC
  - GraphQL

Choix pour ce module

- VueJS
  - Simple et facile à apprendre
  - Permet des projets d'ampleur
  - Reprend des aspects de React et Angular
- Symfony
  - Très répandu
  - PHP = 80% des applications serveurs en 201X

Plan

- VueJS 2
- Symfony 4
- VueJS + Symfony
- Projet

Prérequis

- HTML5
- CSS3
- MySQL
- PHP5
- Programmation Orientée Objet
- Commandes shell Linux
- Modèle client-serveur

Javascript

Historique

- Créé en 1995
- Longtemps relégué à un usage secondaire
  - Ex : script pour vérifier le remplissage d'un champ
- Devenu majeur pour l'interactivité du web
  - Popularisé par réseaux sociaux (Facebook, Twitter …)
  - Eviter le rechargement des pages
- Développement devenu plus simple et plus rapide grâce aux frameworks
  - jQuery (2006), AngularJS (2010), Bootstrap (2011) …

Quelques frameworks

- **Front-end** (navigateur web)
  - 2006 : **jQuery** (John Resig)
  - 2010 : **Backbone.js** (Jeremy Ashkenas)
  - 2010 : **AngularJS** (Google)
  - 2011 : **Bootstrap** (Otto et Thornton, Twitter)
  - 2016 : **Angular 2+** (Google)
    - TypeScript => Javascript
  - 2013 : **React.js** (Facebook)
  - 2014 : **Vue.js** (Evan You, ancien de Google)
- **Back-end** (serveur)
  - **Node.js** : environnement d'exécution ! (≠ framework)

Attention !

- Les frameworks facilitent le développement
  - Fonctionnalités déjà conçues et testées
  - Simplification des développements
  - Accélération des développements
- Les frameworks nécessitent cependant de maîtriser le langage sur lequel ils s'appuient !

Plan

- Notions de bases de Javascript
- Javascript dans les pages web
- Notions avancées
  - polyfills, wrappers, asynchrone, closures
- Echanges de données avec Ajax
- Javascript et HTML5

Norme

- Important pour assurer la compatibilité et l'exécution sur les différents navigateurs web
- ECMAScript (ECMA-262)
  - 1997 : **ES1**
  - 1998 : **ES2**
  - 1999 : **ES3** : try / catch
  - 2009 : **ES5** : support du format JSON
  - 2015 : **ES6** (ES2015) : arrow functions, promises …
  - 2016 : **ES7** (ES2016)
  - 2017 : **ES8** (ES2017) : async / await
  - 2018 : **ES9** (ES2018)

*Standardisé par **Ecma International** (European association for standardizing information and communication systems, créée en 1961, Genève, Suisse)*

Versions de Javascript

- Ne pas confondre la norme avec les versions **réellement** (ou **partiellement**) implantées par les navigateurs web
  - 1996 : **JS 1.0**
  - 1996 : **JS 1.1**
  - 1997 : **JS 1.2**
  - 1998 : **JS 1.3** (ES1, ES2)
  - 1999 : **JS 1.4**
  - 2000 : **JS 1.5** (ES3)
  - 2005 : **JS 1.6**
  - 2006 : **JS 1.7**
  - 2008 : **JS 1.8**
  - 2009 : **JS 1.8.1**
  - 2009 : **JS 1.8.2**
  - 2010 : **JS 1.8.5** (ES5)

JS 1.8.5 (ES5)	Navigateur
Sep. 2012	Chrome 23
Sep. 2012	IE 10 / Edge
Juil. 2012	Safari 6
Avr. 2013	Firefox 21
Juil. 2013	Opera 15

ES6	Navigateur
Août 2016	Edge 14
Sep. 2016	Safari 10
Avr. 2017	Chrome 58
Juin 2017	Firefox 54
Août 2017	Opera 55

ES7 (ES2016)	ES2016+
Chrome 68	Chrome 70
Opera 57	Opera 57
Safari 12	90%
Firefox 63	78%
Edge 18	58%

Moteurs Javascripts

- Les fonctionnalités JS sont apportés par le moteur JS utilisé par le navigateur web
  - **Chakra** : Edge
  - **SpiderMonkey** : Firefox
  - **Chrome V8** : Chrome, Opera
  - **JavaScriptCore** : Safari

Matrices des fonctionnalités supportées par les navigateurs :
ES5 : https://kangax.github.io/compat-table/es5/
ES6 : https://kangax.github.io/compat-table/es6/
ES7 : https://kangax.github.io/compat-table/es2016plus/

$\Rightarrow$ Assurer la comptabilité de son code !!!

$\Rightarrow$ Intérêt des frameworks et transpileurs (Babel : JS $\rightarrow$ JS)

Les bases de JS

Scripts JS dans les pages web

- Vanilla JS

Notions avancées

- Fonctions anonymes
- polyfills, wrappers, asynchrone, closures

Echanges de données avec Ajax

- XMLHttpRequest
- A l'origine des SPA
- websockets

Javascript et HTML5

- Cf. cours de 1A

VueJS 2

Historique

https://github.com/vuejs/vue/releases

- Version 1.0 : octobre 2014
- Version 2.0 : septembre 2016
  - 2.1 : novembre 2016
  - 2.2 : février 2017
  - 2.3 : avril 2017
  - 2.4 : juillet 2017
  - 2.5 : octobre 2017
  - 2.6 : février 2019
    - 2.6.8 => TP (latest stable)
- Version 3.0
  - 2019 ?

https://medium.com/the-vue-point/plans-for-the-next-iteration-of-vue-js-777ffea6fabf

Plan

- Props
- Slots
- Directives
- Scoped styles
- Animations, transitions
- Vue CLI
- VueX

Avantages

- Mélanger HTML et Javascript

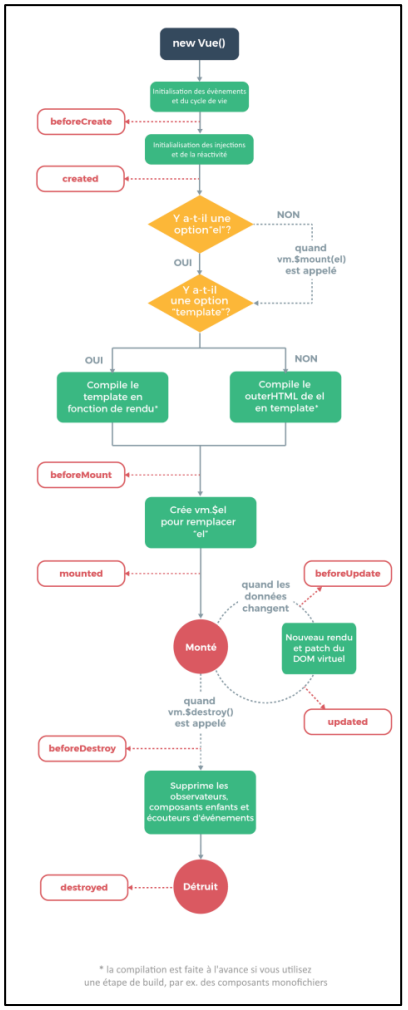
Directives

- Reactive Directives
  - v-text
  - v-html
  - v-show
  - v-class
  - v-attr
  - v-style
  - v-on
  - v-model
  - v-if
  - v-repeat
- Literal Directives
  - v-transition
  - v-ref
  - v-el
- Empty Directives
  - v-pre
  - v-cloak

https://012.vuejs.org/guide/directives.html
https://012.vuejs.org/api/directives.html
https://012.vuejs.org/api/directives.html#Literal_Directives

Cycle de vie

- new Vue()
  - beforeCreate
  - created
  - beforeMount
  - mounted
    - beforeUpdate
    - updated
  - beforeDestroy
  - destroyed



https://fr.vuejs.org/v2/guide/instance.html

PHP 7

- Utiliser un framework requière des bases
  - Compléter celles déjà acquises en « Informatisation du SI »
  - Compléter avec des apports de PHP 7
- Déjà acquis
  - Bases du langage
  - Traitement de formulaires
  - Traitement de fichiers
  - Interrogation BDD
- Nouvelles notions
  - Nouveaux opérateurs
  - Apparition du typage
  - Exceptions
  - Orienté objet
  - Espace de noms

Rappels

- Variables : commence par un « $ »
- Extension des fichiers : .php
- Code PHP entre <?php et ?>
- Type de données

- Opérateurs

Rappels

- Syntaxe proche du C
  - if () {} [else if () {}] [else {}]
  - for (; ;) {}
  - while () {}
- function multiplier ($x, $y) {

```
    return $x * $y ;
}
```

La boucle foreach()

- Très utile pour parcourir les tableaux associatifs

Historique

- PHP5
- PHP7

Symfony 4

https://symfony.com

Cadriciel (Framework)

- Objectifs
  - Ne pas réinventer la roue à chaque projet
  - Partager les bonne pratiques (structurant)
  - Offrir un « cadre de travail »
- Fournir des composants **aux développeurs**
  - Accélérer la création des fondations, de l'architecture et des grandes lignes d'un logiciel
  - **Pas** destiné aux utilisateurs finaux (cf. Wordpress)
  - **Pas** destiné aux novices

- Intérêts
  - Se recentrer sur le fond, plutôt que sur la forme
- Avantages
  - Accroître la productivité des développeurs
  - Aspects structurant : organisation du code
    - Favorise évolutivité et maintenabilité
  - Briques déjà testées, éprouvées et optimisées
  - Recruter collaborateurs partageant les pratiques
  - Communauté de développeurs
    - Documentation, aide, résolution de bugs, mises à jour

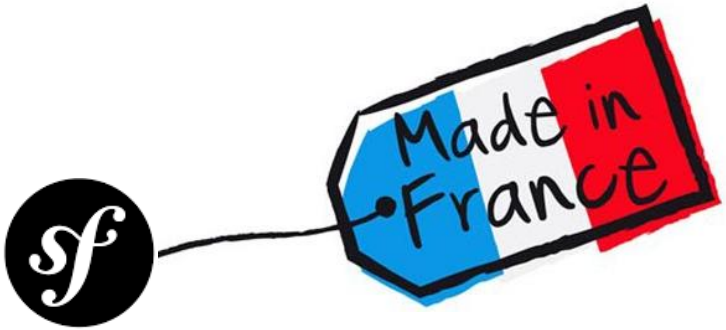
- Inconvénients
  - Courbe d'apprentissage plus élevée
    - Temps d'apprentissage non négligeable
    - Connaître chaque brique du framework prend du temps
    - Apprendre un framework est un investissement personnel
  - Se tenir au courant des nouveautés
    - Changements lors des nouvelles versions
    - Suivit de l'évolution des langages

Quelques frameworks PHP

- Symfony
- Laravel (construit à 30% sur Symfony)
- CodeIgniter
- Zend Framework
- CakePHP
- Yii
- Phalcon
- FuelPHP

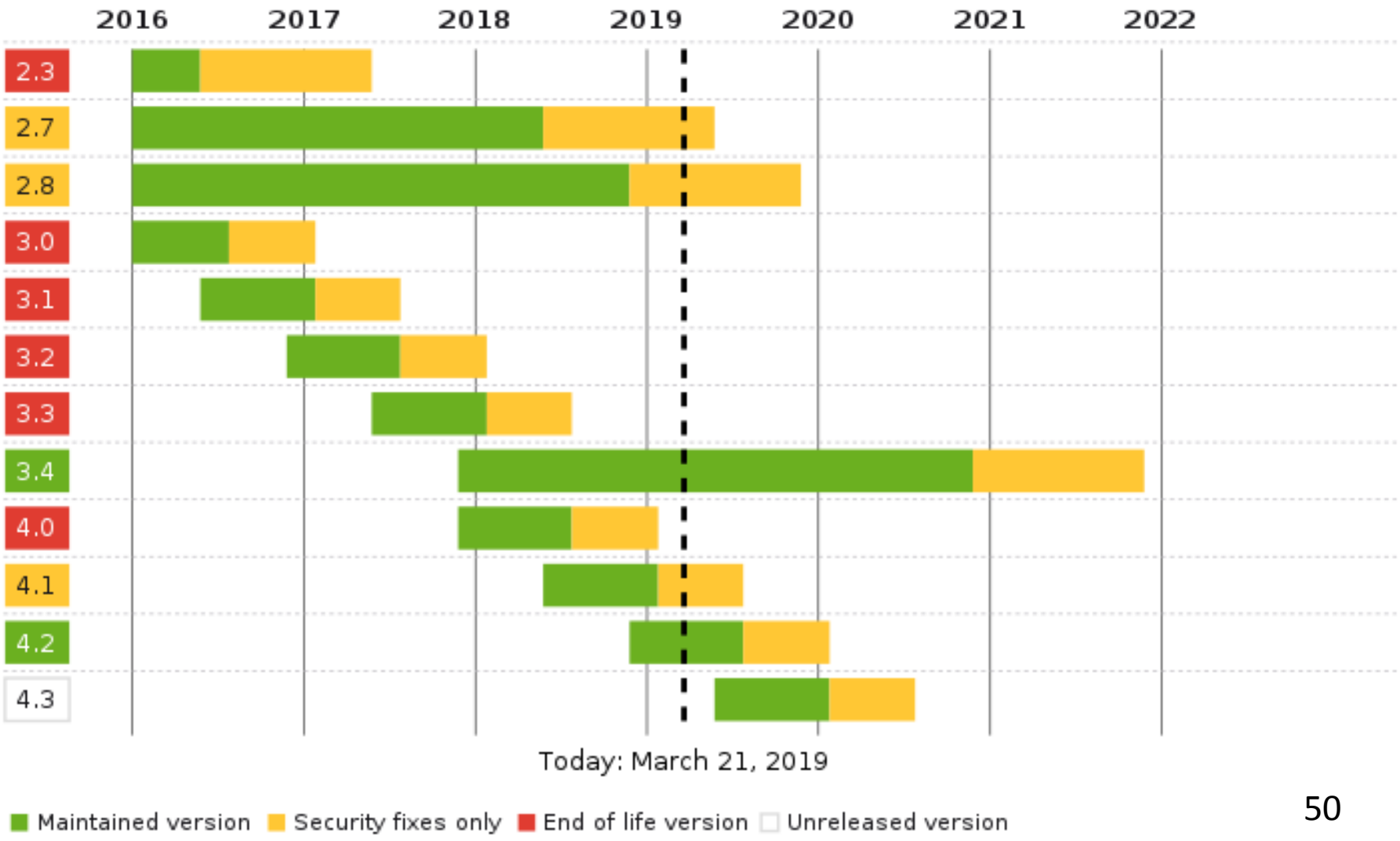
Historique

- Agence SensioLabs
  - **S**ensio **F**ramework
  - Créateur : Fabien Potencier
- Symfony 1 : octobre 2005
- Symfony 2 : août 2011
- Symfony 3 : novembre 2015
- Symfony 4 : novembre 2017 (PHP > 7.1)
  - 4.1 : mai 2018
  - **4.2 : novembre 2018** (PHP ≥ 7.3)
    - 4.2.4 => TP (latest stable)
  - 4.3 : mai 2019



Versions

https://symfony.com/roadmap



Plan

- Twig
- Doctrine ORM
- API Platform
- Architecture MVC
- Open Source
  - Licence MIT

Créer un projet

- En utilisant le gestionnaire de dépendances
  - composer create-project symfony/skeleton mon-projet
- Architecture des fichiers
  - bin/ : programmes utiles pendant le développement
  - config/ :
  - public/ : fichiers destinés aux visiteurs
    - images, fichiers CSS, codes Javascript
    - index.php : le contrôleur frontal (point d'entrée de l'application : prévient le noyau de l'arrivée d'une requête)
  - src/ : code source du projet
    - Organisation du code en « bundles »
    - Kernel.php : gestion des requêtes, lancement des codes à exécuter
  - templates/ : les fichiers twig pour le moteur de templating
  - var/ : utilisé pendant l'activité du projet
    - fichiers de logs, fichiers de cache ...
  - vendor/ : contient les bibliothèques externes au projet
    - Y compris celles de Symfony, mais aussi Doctrine, Twig …

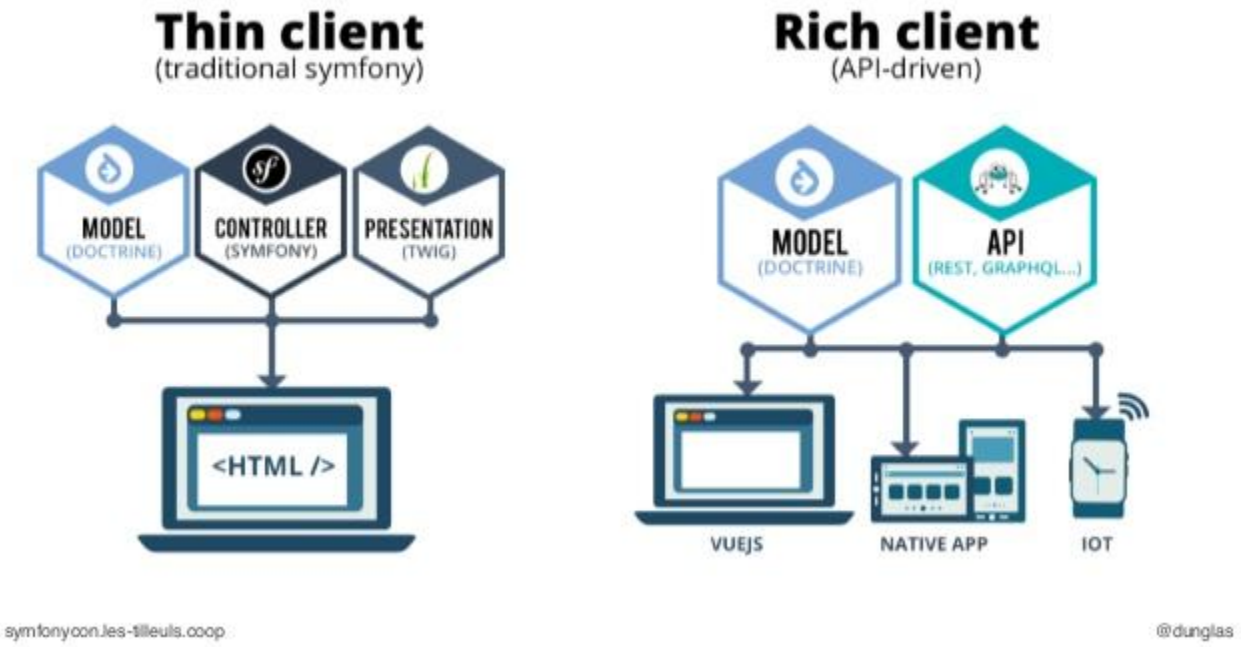
Documentation

- Synfony API
  - https://api.symfony.com/4.2/index.html

Références bibliographiques

VueJS + Symfony

Deux possibilités



Docker

Ce que nous n’avons pas vu

- Authentification des utilisateurs
- Internationalisation de l’application
- Environnement de tests unitaires (PHPUnit)

Bibliographie

- **The Majesty of Vue.js** (Packt Publishing, 2016, 230 pages) scholarvox.com/book/88843467
- **Développez votre site web avec le framework Symfony3** (Eyrolles, 2016, 537 pages) scholarvox.com/book/88836319
- **PHP 7 : Cours et exercices** (Eyrolles, 2017, 609 pages) scholarvox.com/book/88838646
- **Découvrez le langage JavaScript** (Eyrolles, 2017, 504 pages) scholarvox.com/book/88838415