Drupagora 2019

L'effervescence des frameworks front-end **réactifs**

du design aux composants web performants

Hello!

Je suis Rodolphe

Co-fondateur d' alsacréations

SOMMAIRE

1. Contexte historique
2. Écosystème Vue.js
3. Design de composants
4. Headless, SPA, code & co.
5. Impact sur le(s) métier(s)

Il était une fois...

* dates simplifiées

Vue.js

- Framework JavaScript
- Model-view-ViewModel (MVVM)
- Excellente documentation (fr)
- Progressif : barrière d'entrée faible
- Ensemble cohérent : vue-router, vuex, etc.
- 140 000 ★ sur GitHub (a dépassé React)

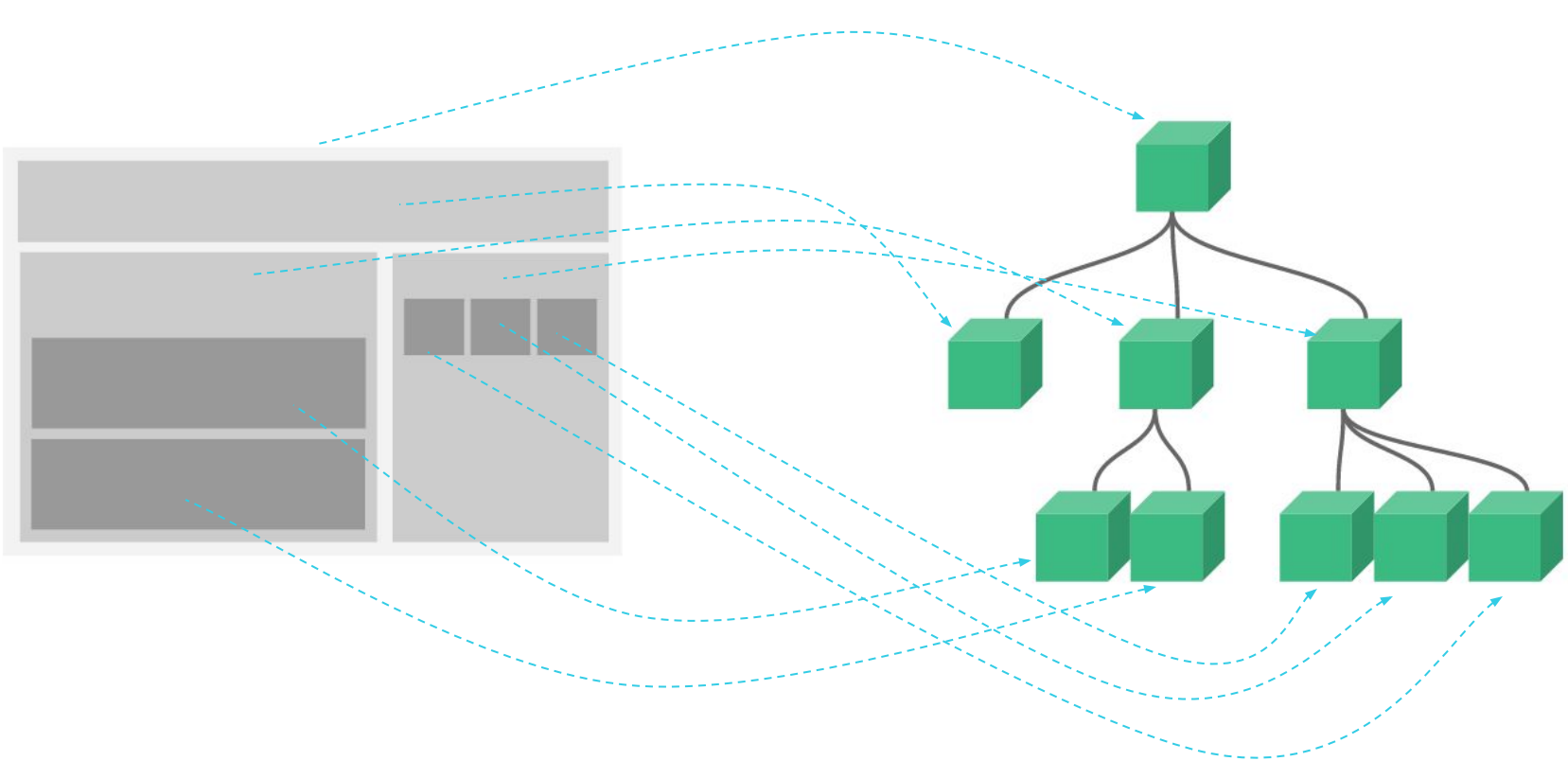
892 627

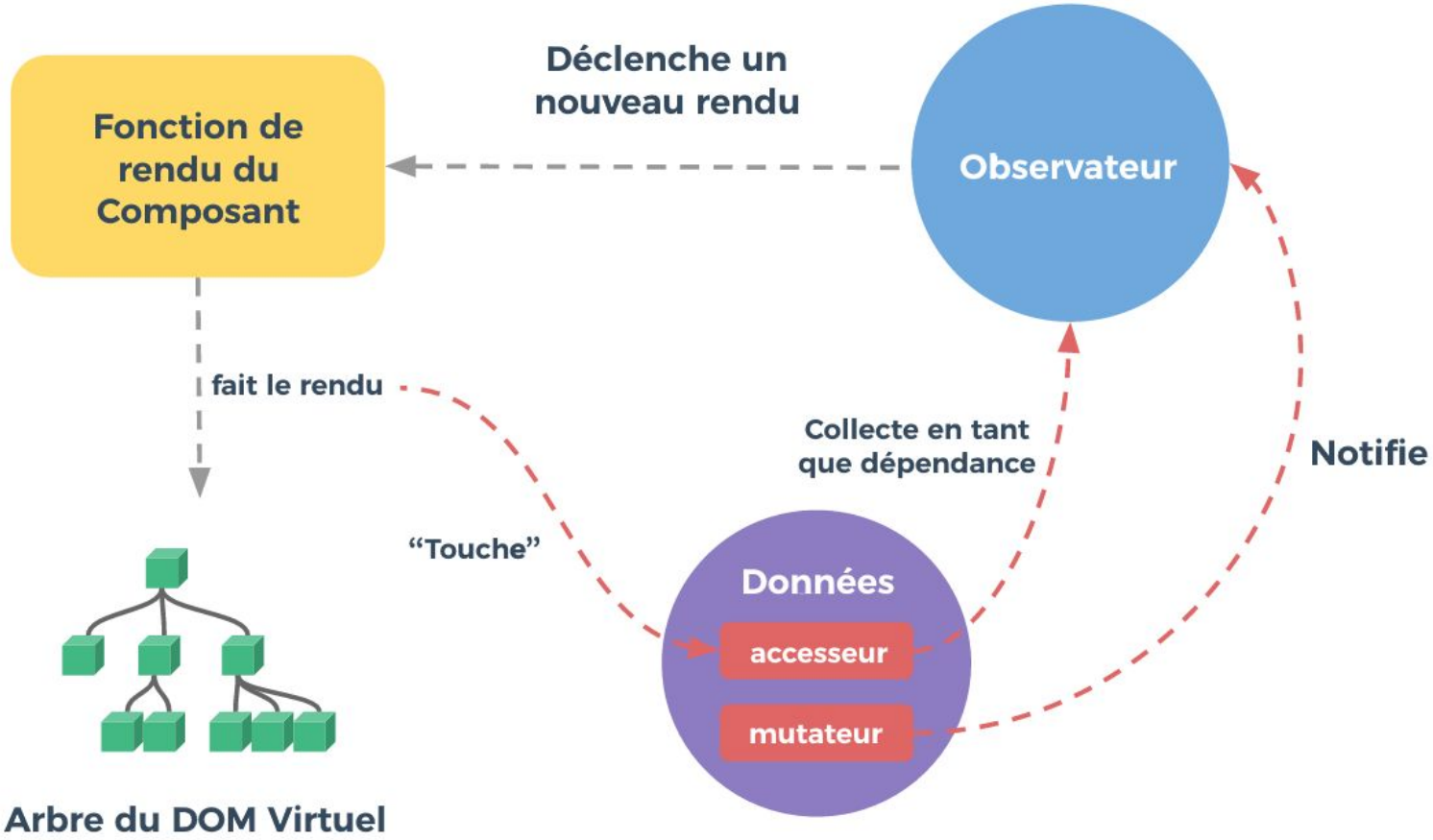
Projets utilisent Vue.js sur GitHub


1
R
React
2
V
Vue.js
3
Ng
Angular
9.2%
19.1%
64.8%
20.5%
46.6%
28.8%
31.8%
10.4%
33.8%
23.9%
React
Vue.js
Angular
Pas intéressé
Je voudrais l'apprendre
Je ne l'utiliserai plus
Je l'utiliserai encore

Vue d'ensemble

- Templating réactif
- Données locales (*store*) = plus grande réactivité
  - Réseau : se mesure en fractions de secondes
  - Local : se mesure en millisecondes ou moins
- Routage : c'est le front-end qui gère les URL et non plus le CMS back-end
- Requêtes d'API, synchronisation des données
- Développement continu avec live reloading, etc.




Déclenche un
nouveau rendu
Fonction de
rendu du
Composant
Observateur
fait le rendu
Collecte en tant
que dépendance
Notifie
"Touche"
Données
accesseur
mutateur
Arbre du DOM Virtuel

Design et code

des maquettes aux composants front-end

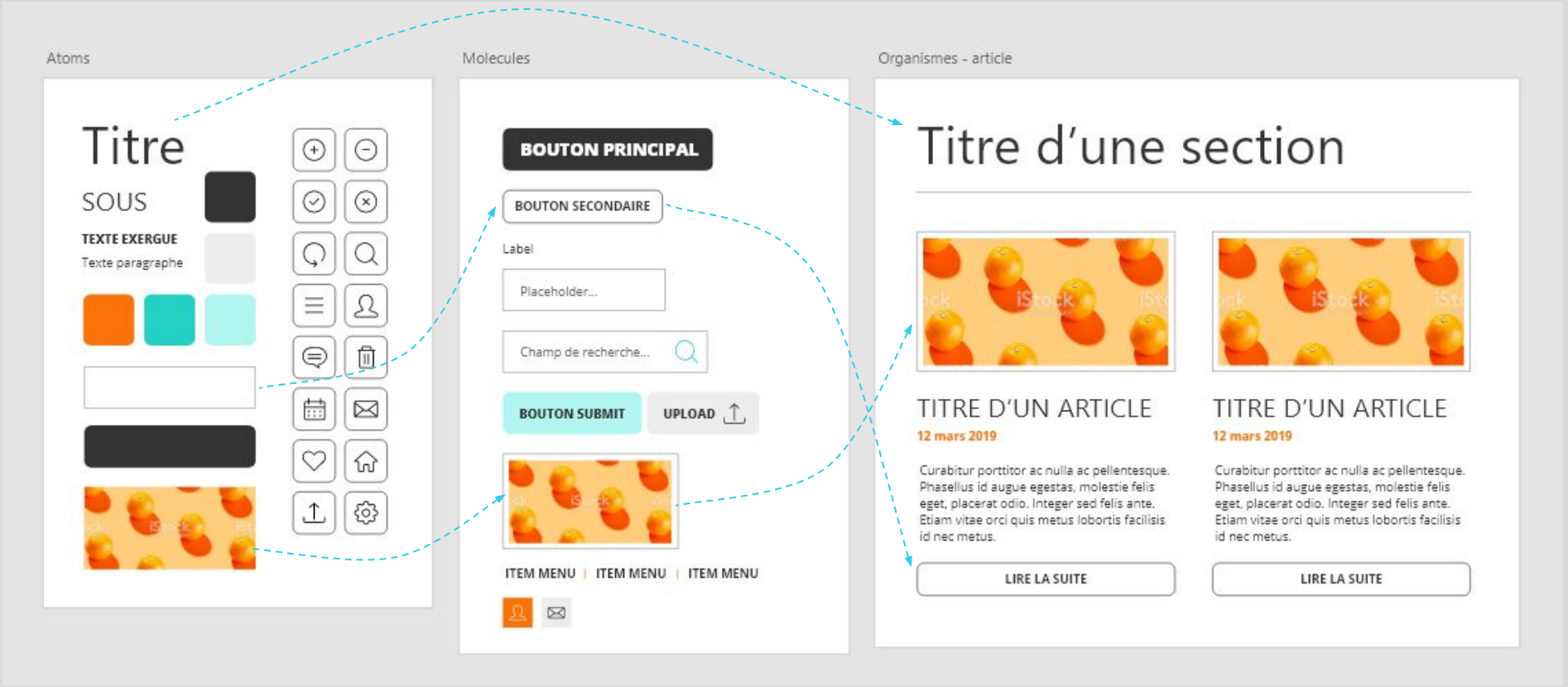
Design

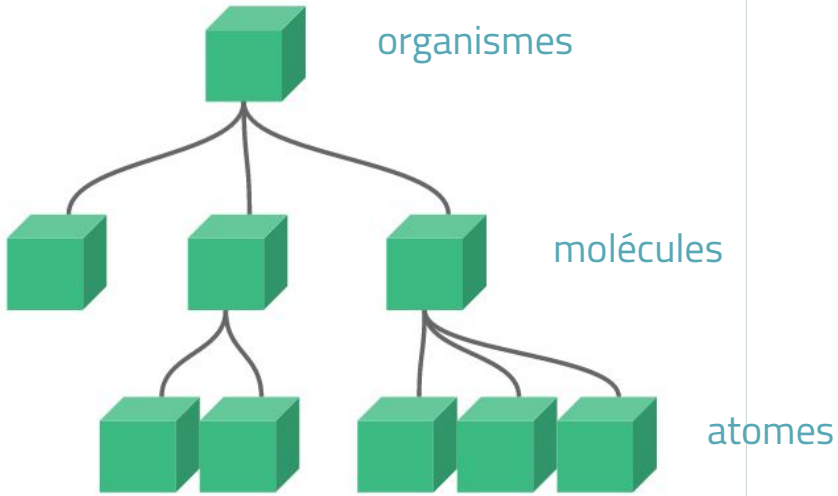
- Atomic design
- Design system

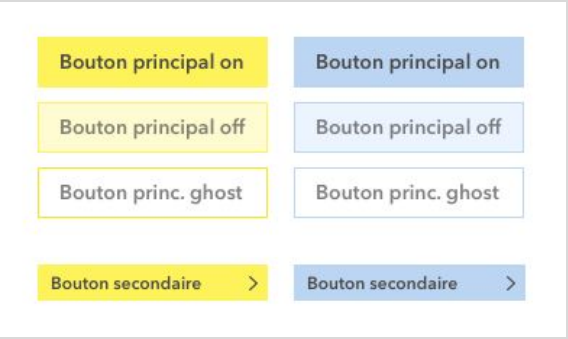
≈ Design* de composants

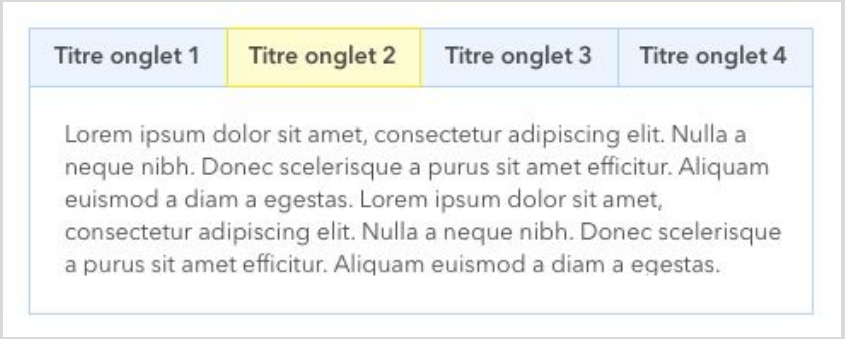
* conception graphique mais aussi fonctionnelle

Atomic design

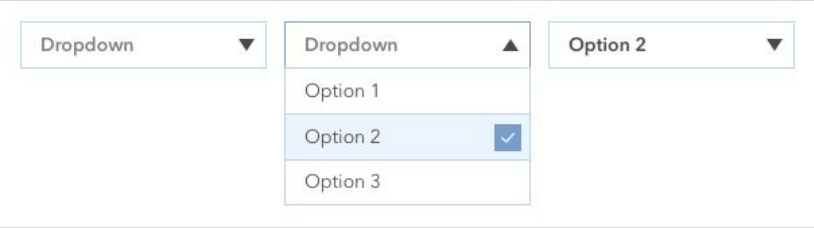



organismes
molécules
atomes





Design system



GOV.UK uses cookies to make the site simpler. Find out more about cookies

GOV.UK Design System

Search Design System

BETA This is a new service – your feedback will help us to improve it.

Get started Styles Components Patterns Community

Design your service using GOV.UK styles, components and patterns

Use this design system to make your service consistent with GOV.UK. Learn from the research and experience of other service teams and avoid repeating work that's already been done.

Get started >

Styles

Make your service look like GOV.UK with guides for applying layout, typography, colour and images.

Browse styles

Components

Save time with reusable, accessible components for forms, navigation, panels, tables and more.

Browse components

Patterns

Help users complete common tasks like entering names and addresses, filling in forms and creating accounts.

Browse patterns

Community

Design System beta

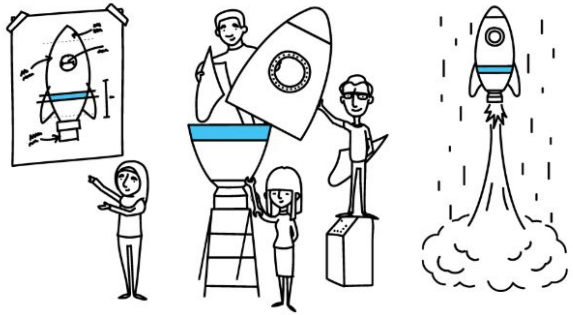
Uniting teams that create digital services, through inclusive design, open-source code and shared insights.

About

The Australian Government Design System provides a framework and a set of tools to help designers and developers build government products and services more easily.

The system incorporates the highest usability and accessibility standards and helps deliver a consistent experience for all users, in line with the Digital Service Standard.

Get started



Search...

Getting started
Layout
Content
Components

Utilities
Borders
Clearfix
Close icon
Colors
Display
Embed
Flex
Float
Image replacement
Position
Screenreaders
Sizing
Spacing
Text
Vertical align
Visibility

Extend
Migration
About

COLORS

Convey meaning through color with a handful of color utility classes. Includes support for styling links with hover states, too.

COLOR

Contextual text classes also work well on anchors with the provided hover and focus states. **Note that the `.text-white` and `.text-muted` class has no link styling.**

By theme

.text-primary

.text-secondary

.text-success

.text-danger

.text-warning

.text-info

.text-light

.text-dark

.text-muted

.text-white

Copy

```
<p class="text-primary">.text-primary</p>
<p class="text-secondary">.text-secondary</p>
<p class="text-success">.text-success</p>
<p class="text-danger">.text-danger</p>
```

By color name

.text-blue

.text-purple

.text-pink

.text-red

.text-orange

.text-yellow

.text-green

.text-teal

.text-cyan

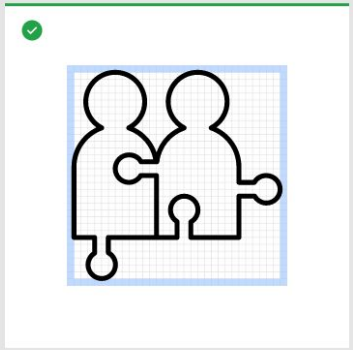
.text-white

.text-gray

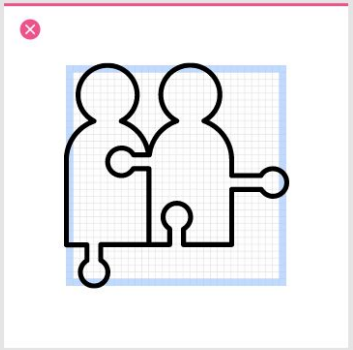
.text-gray-dark

Muted link

Color
Background color



Do treat padding as a safe zone for overflowing strokes.



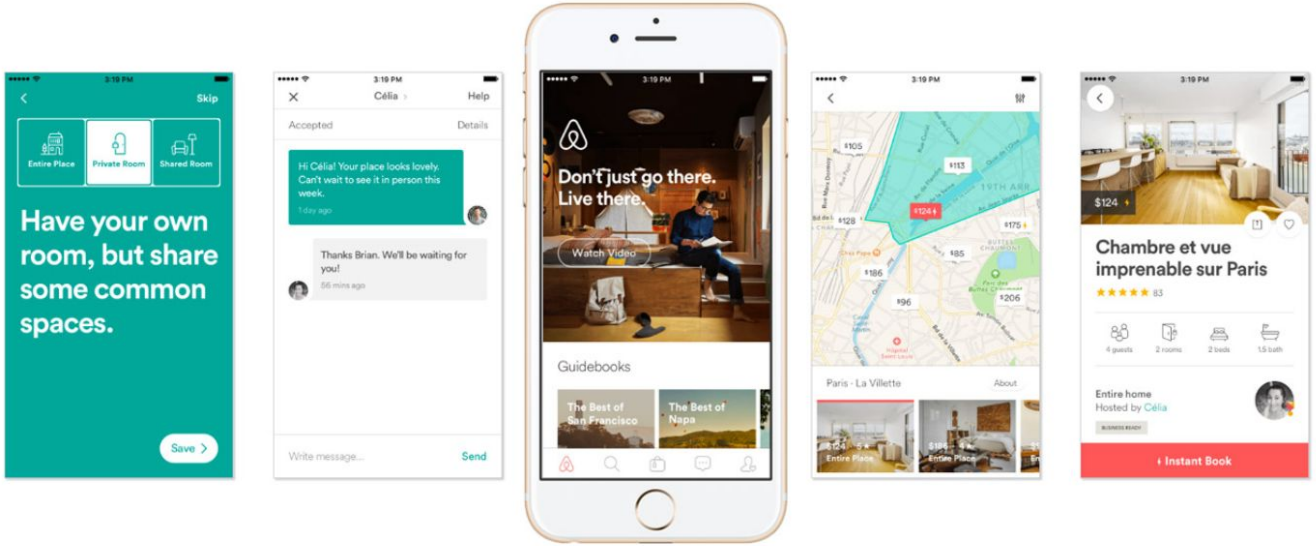
Don't place design elements on the outer edges of the safe zone.

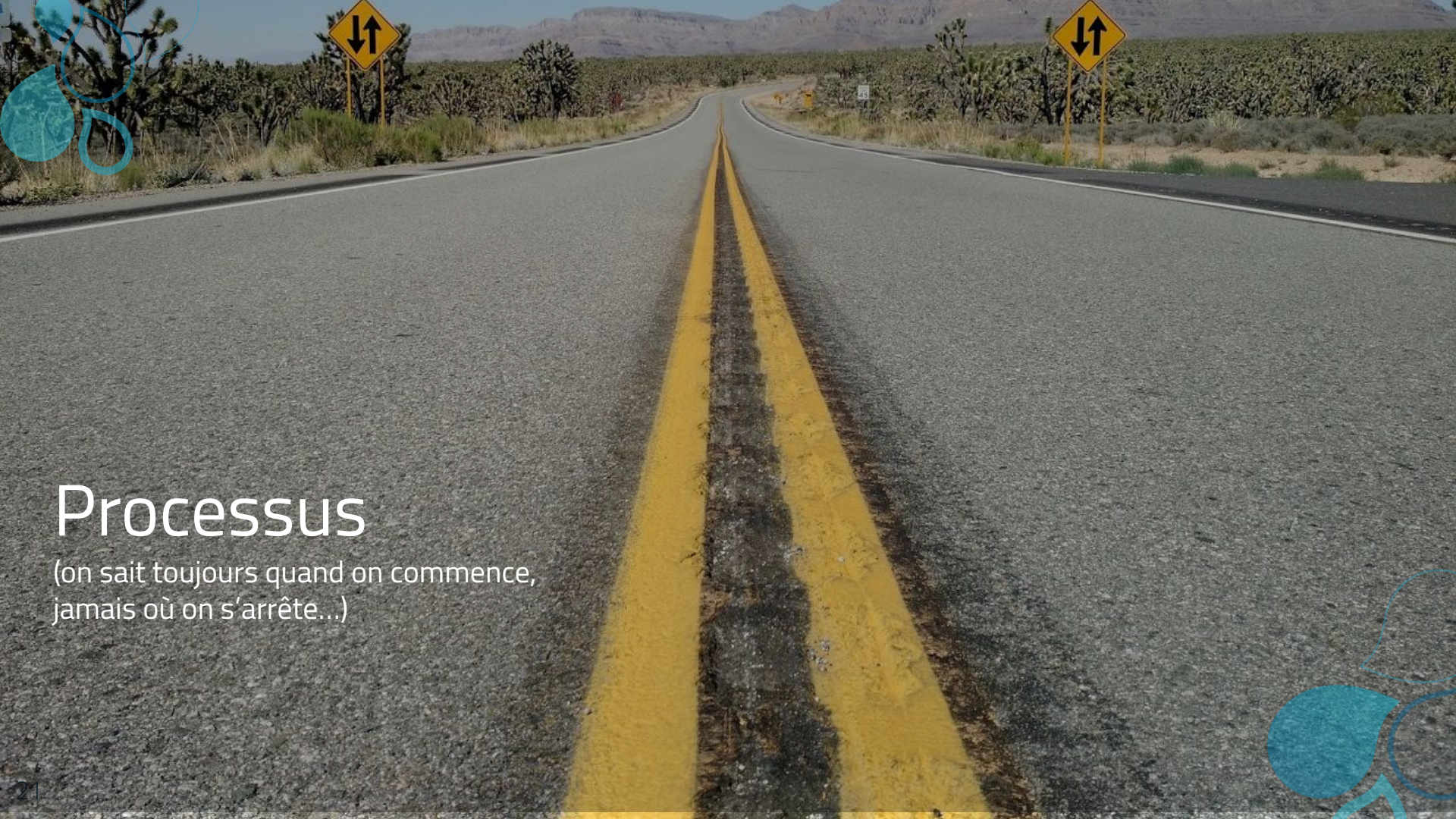
Style

The stylistic conventions of IBM's pictograms deliver meaningful metaphor through simple line art. Each symbol is intentionally designed to harmoniously pair with IBM Plex™. The juxtaposition of smooth curves and sharp angles is central to IBM Design.

Lessons Learned

We knew that this was a challenging project. It meant re-designing and rebuilding the majority of the views in our app. We managed to make our goal of creating the system and releasing the new apps on April 17th. As with any project, there are things we wish we would have done differently.





Processus

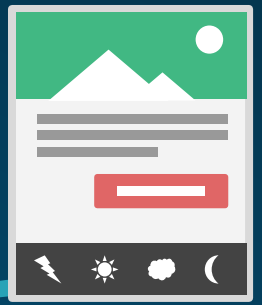
(on sait toujours quand on commence, jamais où on s'arrête…)

Processus

Ergonomie

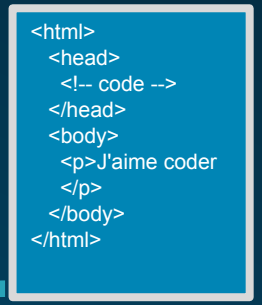
Tests utilisateurs (prototype interactif)

Design



Tests utilisateurs (prototype interactif)

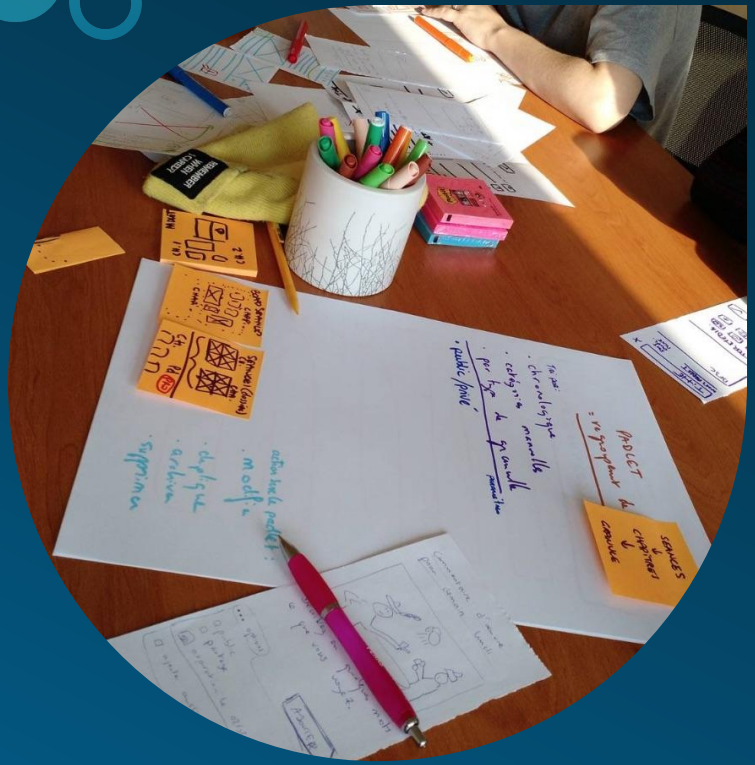
Intégration



Composants

Vue.js

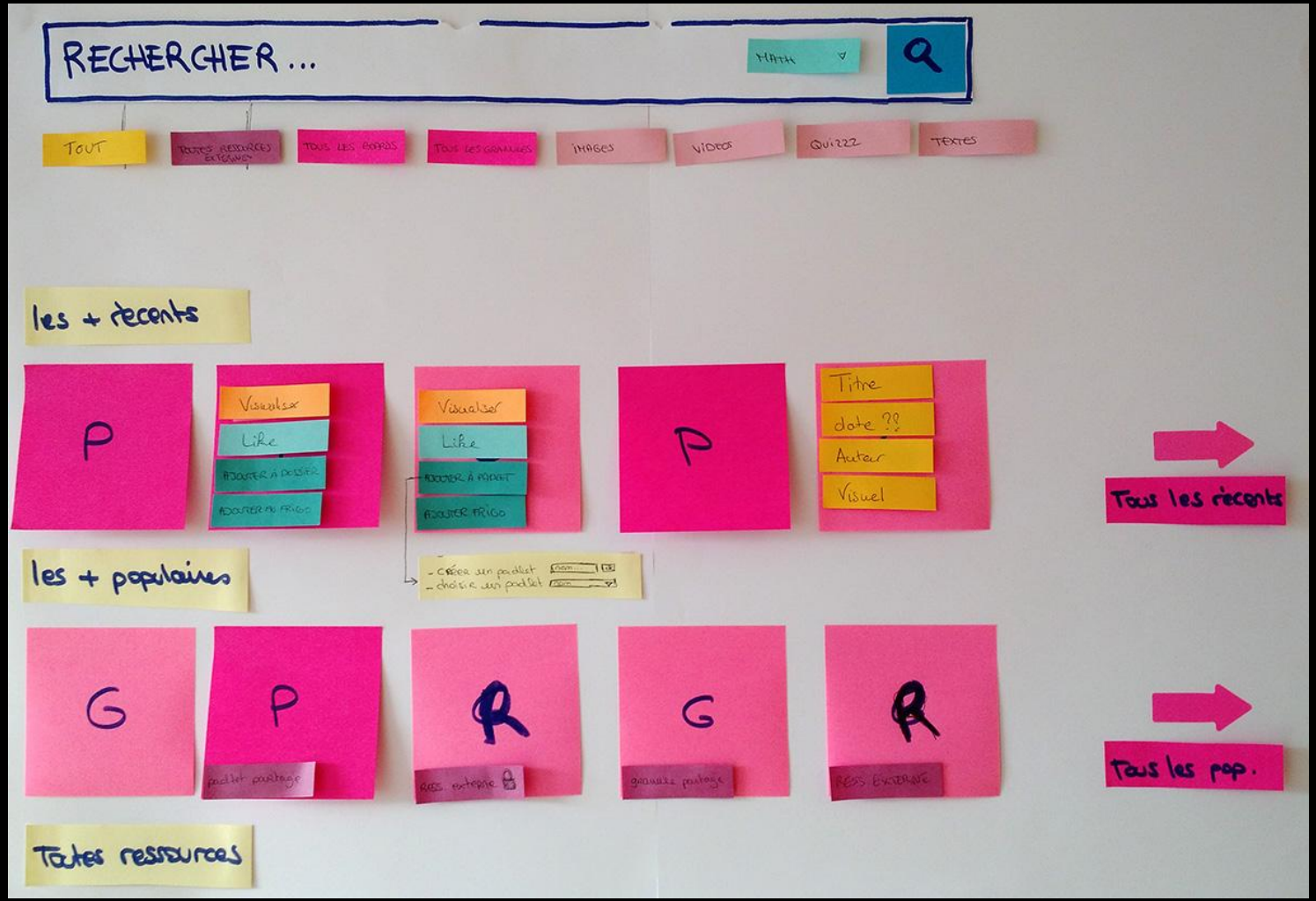
Design system



Élaboration

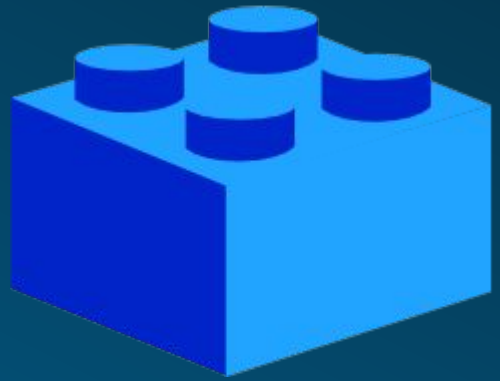
- Spécifications
- Design
- Intégration
- Tests et retours
  - Par composant
  - Par imbrication de composants
  - Par vue
  - Tests unitaires


like produit
création
mot cle de recherche?
date supp / modif?
LAYOUT
OPT. GRAPH.
public / privé
partage
TITRE + DESC
SUIVANT
AJOUT D'UNI GRANULE
INDEX.
SUPPR / MODIF
DUPLIQUER
MODIFIER
INDEXER
AJOUTER
choisir un template prefait
TITRE
description
auteur
like dupliquer partager suivre
TIT
DEPLACER
auteur
LIKES (nbre)
AIMANTÉ (côté)
MODIFIER
TITRE
ETIQUETTES :
ANNULER
MODIFIER


RECHERCHER ...
MATH
TOUT
TOUTES RESSOURCES EXTERNES
TOUS LES POSTS
TOUS LES GROUPES
IMAGES
VIDEOS
QUIZZZ
TEXTES
les + récents
P
Visualiser
Like
AJOUTER À PADLET
AJOUTER AU FRIGO
Visualiser
Like
AJOUTER À PADLET
AJOUTER FRIGO
- créer un padlet
- choisir un padlet
P
Titre
date ??
Auteur
Visuel
Tous les récents
les + populaires
G
P
padlet partagé
R
RES externe
G
groupe partagé
R
RES EXTERNE
Tous les pop.
Toutes ressources

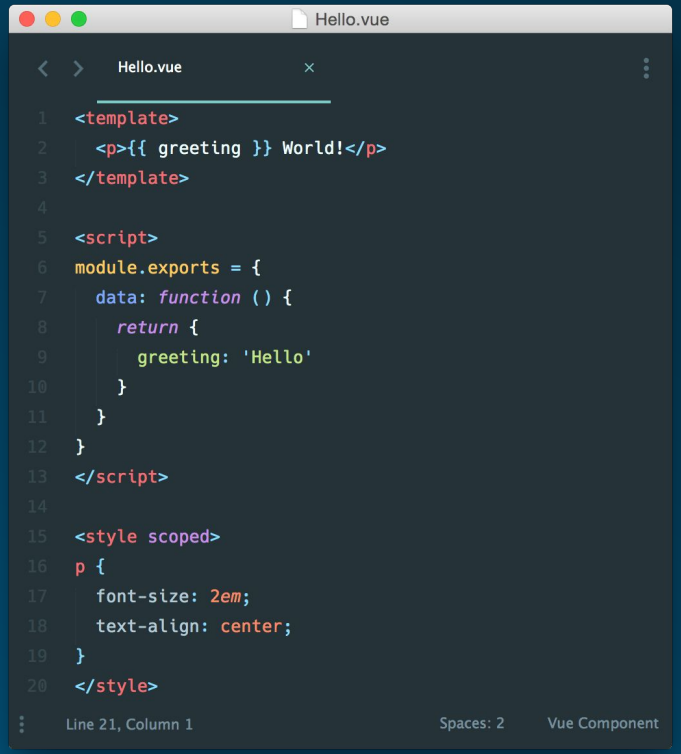

Un immense lego©
de la brique à l'application complète

Brique brique brique



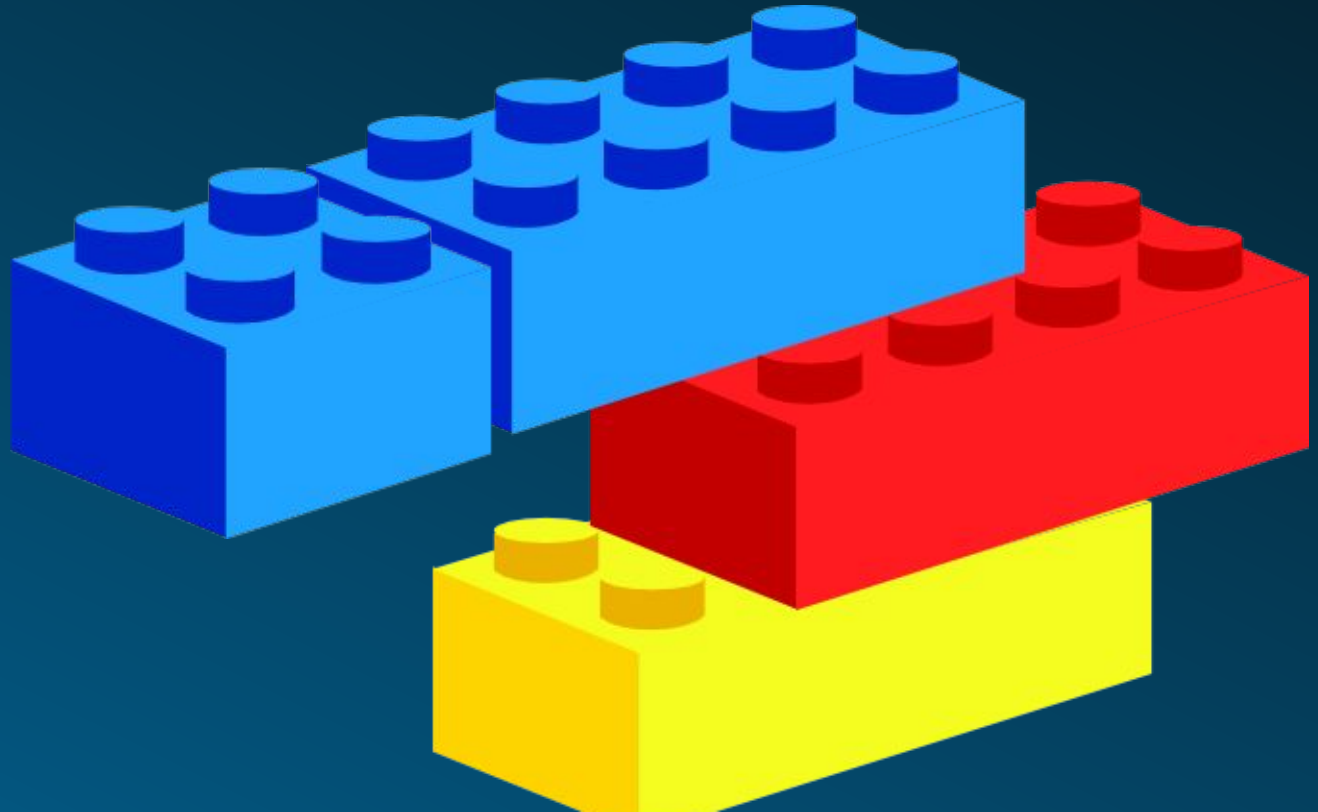
- Se suffit à elle-même
- Est réutilisable
- Dispose de ses propres données
- Peut s’interfacer avec d’autres composants

Brique brique brique



- Se suffit à elle-même
- Est réutilisable
- Dispose de ses propres données
- Peut s'interfacer avec d'autres composants

Brique brique brique



Élimine le dead code

- Les projets ont toujours tendance à grossir
  - Difficulté de maintenance
  - Performances en baisse
- Lorsqu'on retire un composant, tout son code (logique, styles, données) et éliminé
- Tree shaking

Web Components

Spécifications W3C, existe aussi en polyfills et frameworks dédiés (Polymer)

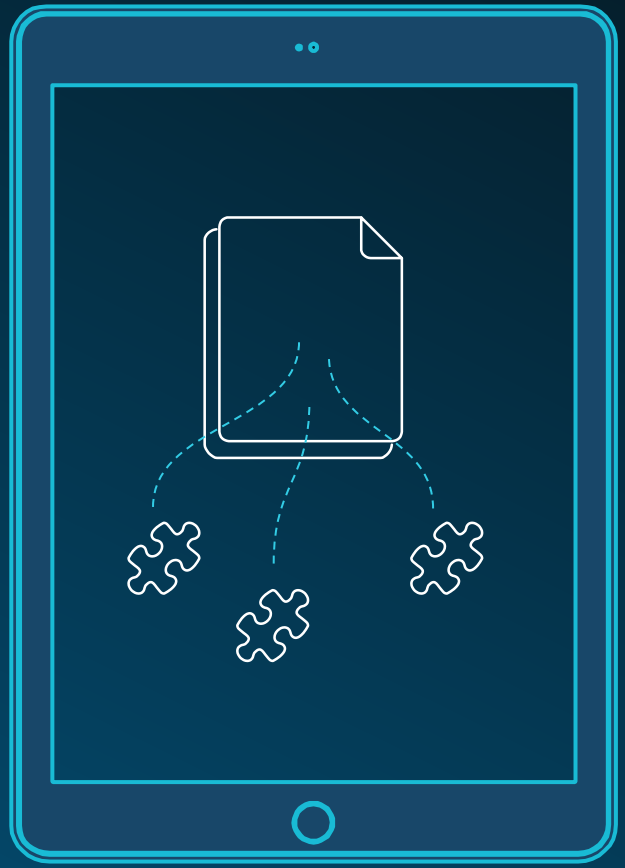
- Custom Elements
  - Pouvoir créer ses propres éléments HTML
- Shadow DOM
  - Pouvoir les “encapsuler” (styles/scripts) cf : <video>
- HTML Template
  - Pouvoir utiliser des portions réutilisables
- ES Module
  - Modules Javascript réutilisables (import/export)


Headless, SPA, API...
de l'inconnu à la tranquilité
.woff

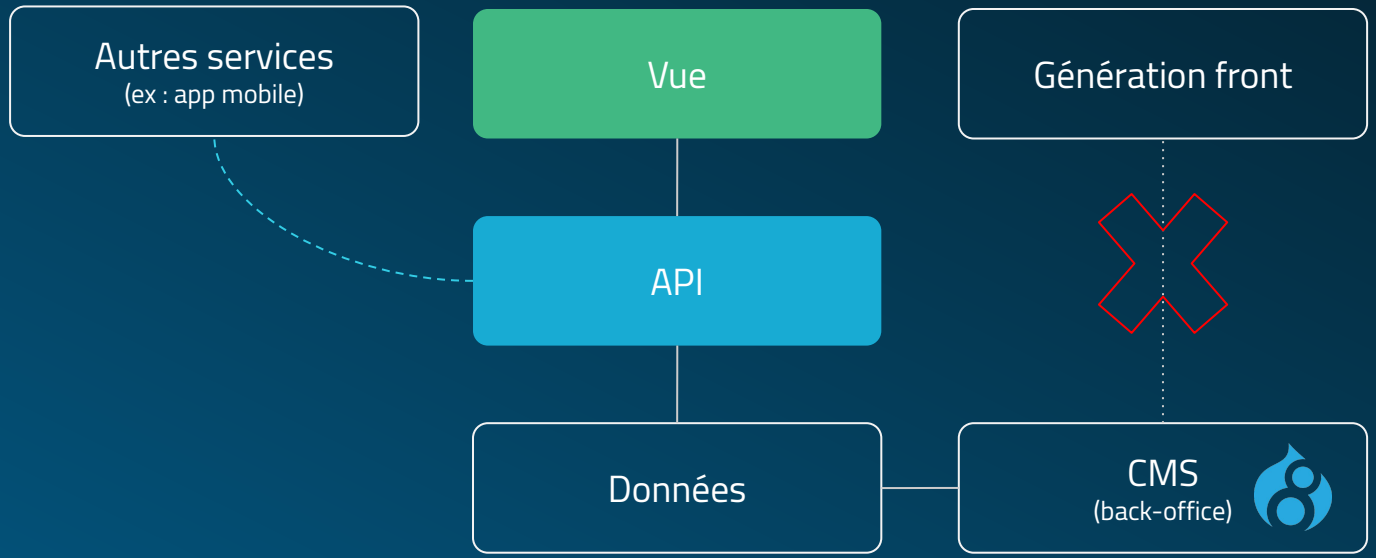
SPA

Single Page Application

On préserve un contexte d'exécution (le même document HTML parent), tout comme dans une application classique, n'obligeant pas à tout recharger depuis le serveur à chaque changement de vue, de lien....



Headless

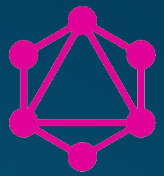

Autres services
(ex : app mobile)
Vue
Génération front
API
Données
CMS
(back-office)

Headless

- Pas de nécessité de générer les pages finales : on délègue au front-end
- Besoin d'API : on est tous happy avec des API
  - Services délivrant de la donnée brute
    - REST
    - GraphQL ? (autre conférence à venir)

GraphQL

- Langage de requêtes pour API proposé par Facebook pour économiser les données sur les réseaux mobiles lents (~ 2012)
- Un seul appel pour obenir un arbre de données
- Ces données *peuvent* être injectées dans les composants
- Utilisable avec les interfaces courantes (Ajax, fetch, axios…)



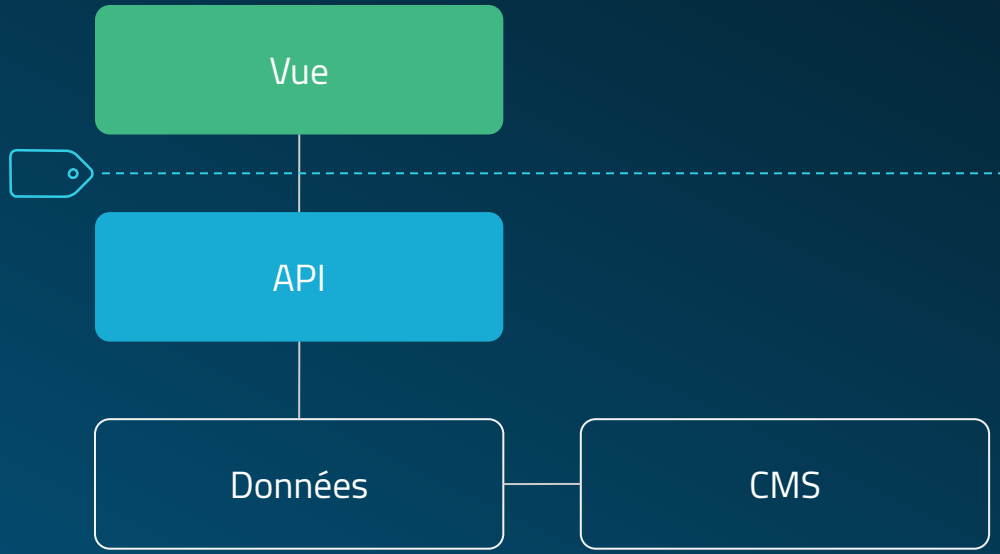
GraphQL

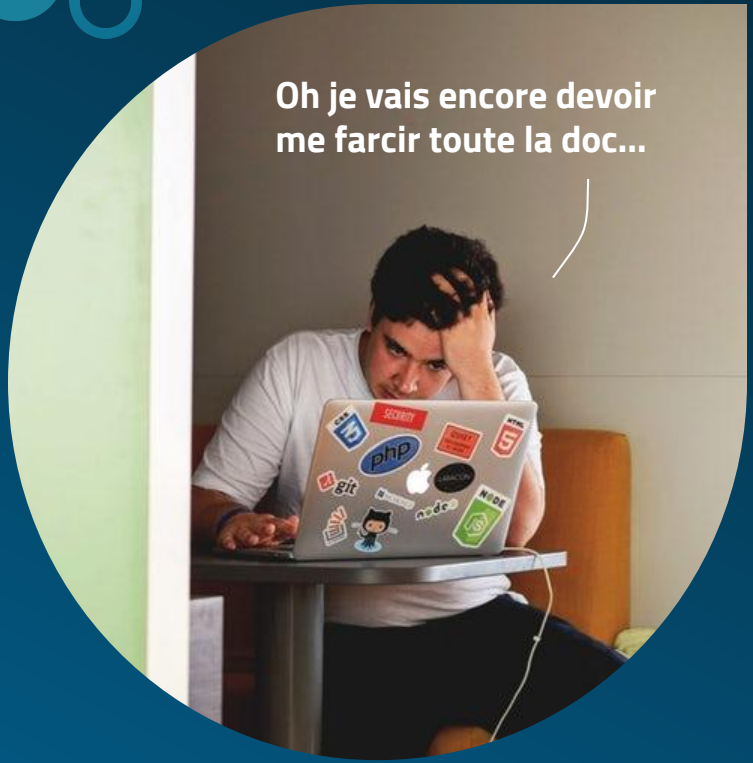
Par le découpage de la conception,
on optimise la **robustesse**,
la **performance**
et les **ressources** métier.

Moi

Coupez !

Dissocier le développement
front-end (par le client ou vous)
du back-end (par le client ou vous)





Cela peut faire peur

- L'inconnu
  - Nouveaux outils, nouvelles méthodes
- Résistance au changement
- Montée en compétence
  - 3 à 6 mois pour devenir "bon"
- Cela vaut-il l'investissement ?

Chacun son <style>

Laissons les profils se concentrer sur leur domaine de prédilection

Développeur
~~Magneto~~
~~Magento~~
Drupal

Back

Données
Sécurité

Front

Réactivité
UI/UX

Performance

Intégratrice
front-end

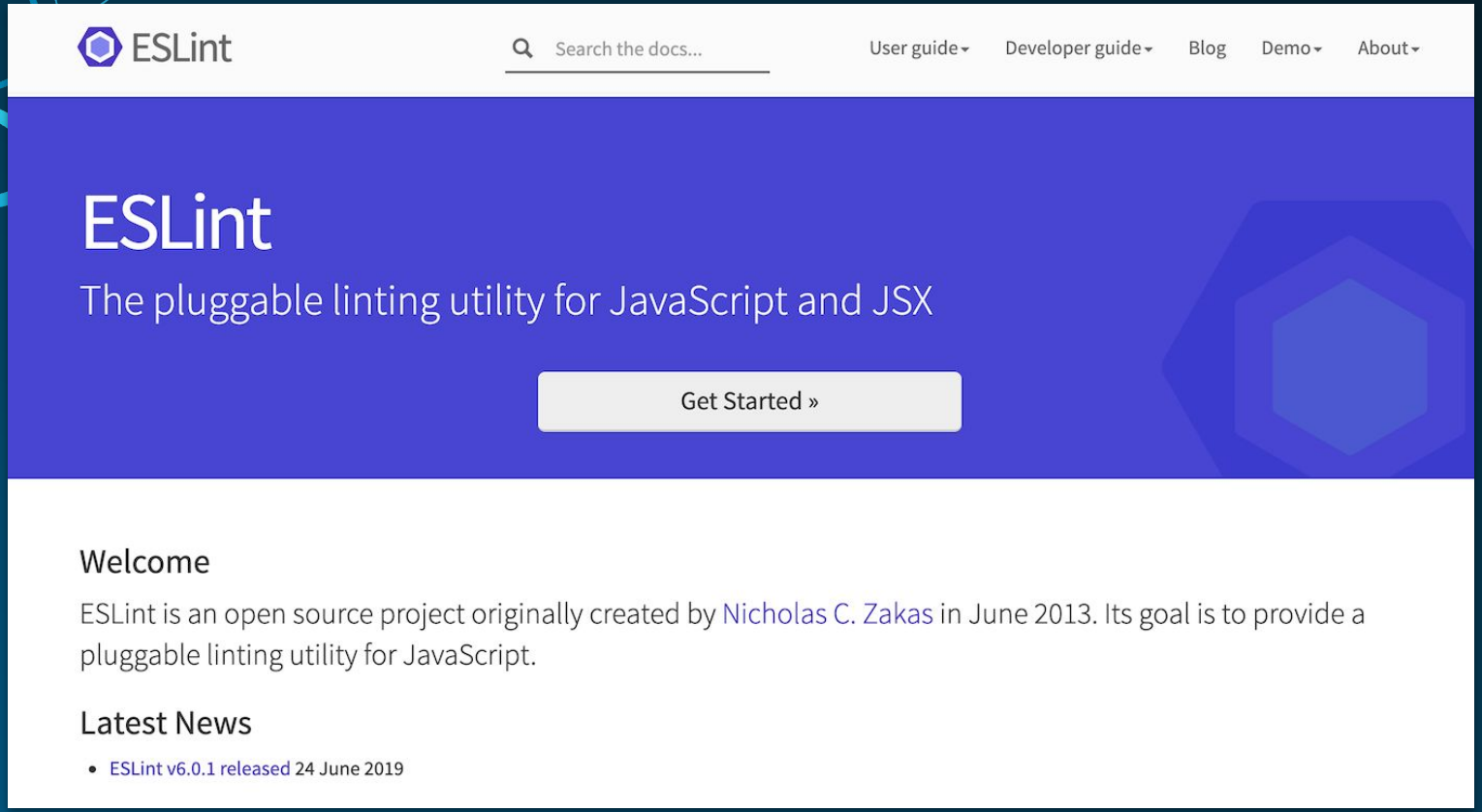
Impact sur le métier

- Repenser les rôles (et le recrutement) avec de nouveaux profils front-end
- Intégration statique + dév. JS synchrones
- Tests avec “données réelles”
- On abandonne totalement le silo par étapes : statique > js > tests
- Obtention MVP plus rapide

Progression

Cycle en tuyau

Agilité



https://eslint.org/

Available rules

Base Rules (Enabling Correct ESLint Parsing)

Enforce all the rules in this category, as well as all higher priority rules, with:

```json
{
  "extends": "plugin:vue/base"
}
```

Rule ID	Description	
vue/comment-directive	support comment-directives in `<template>`	
vue/jsx-uses-vars	prevent variables used in JSX to be marked as unused	

Priority A: Essential (Error Prevention)

Enforce all the rules in this category, as well as all higher priority rules, with:

vue/no-unused-components

Disallow registering components that are not used inside templates

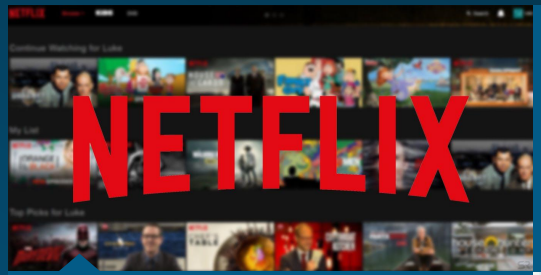
- ⚙️ This rule is included in all of `"plugin:vue/essential"`, `"plugin:vue/strongly-recommended"` and `"plugin:vue/recommended"`.

vue/order-in-components

Enforce order of properties in components

- ⚙️ This rule is included in `"plugin:vue/recommended"`.
- 🔧 The `--fix` option on the command line can automatically fix some of the problems reported by this rule.

Conquis par React ou Vue



Netflix

Interfaces internes de gestion des infrastructures

Airbnb

Applications web et mobiles

Facebook

Newsfeed

Conquis par React ou Vue

Adobe

Adobe Portfolio

Xiaomi

Interfaces utilisateurs

Nintendo

Comptes utilisateurs


De la Silicon Valley
à chez vous

Retour d'expérience Vue/headless

Projet de "réseau social documentaire"

- Performance de chargement d'une vue
  de 1500ms à 50ms
- Quantité de données en transit
  -50% à -90%

Démo ?

On a encore un peu de temps ?

Allons-y.

Recommandations

- Pensez “composants” dès le départ
- Ne négligez pas la qualité du front-end
- Construisez un modèle de données solide
- Partagez les acquis entre développeurs back, front et designers
- Mettez en place des conventions et automatismes de code (eslint) pour le travail en équipe

Merci !

Des questions ?

- Twitter : @diou / @alsacreations
- www.alsacreations.fr

alsacréations

Images : unsplash.com & photos R.R.