L'architecture micro-front,

un levier pour la webperf ?

adeo LEROY MERLIN

Bertrand LAOT

Ludovic LAGATIE

LES GRANDS THÈMES ABORDÉS

01 QUI SOMMES NOUS ?

Nous, nos concurrents, notre historique IT

02 L'ARCHITECTURE MICRO FRONT

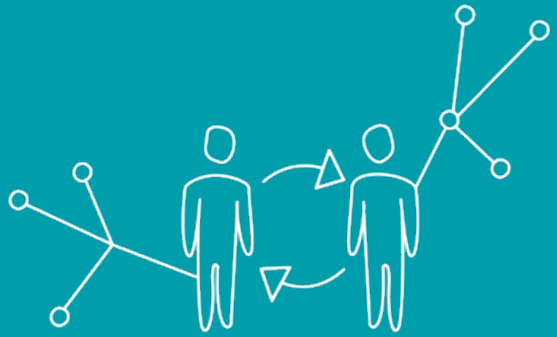
Les principes, les choix techniques

03 PERFORMANCE ET RÉSILIENCE

Performance et résilience côté serveur
Avec ou sans cache ?
Les ressources statiques, les tiers, le CDN
La culture webperf à Leroy Merlin

- **142 magasins en France**
- **Sixième site e-commerce français au premier semestre 2021 et cinquième au deuxième trimestre 2021 (en nombre de visiteurs par mois)** *(source Fevad / Médiamétrie)*
- **24% de la population française visitent notre site tous les mois** *(source Fevad / Médiamétrie)*
- **Une répartition de trafic 55% mobile, 45% desktop** *(source Google Analytics)*
- **Une clientèle et des devices très hétérogènes**
- **7 pages vues en moyenne par visite** *(source Google Analytics)*
- **… et 1er au classement webperf e-commerce JDN / Fasterize**

LES CONCURRENTS



Les click and mortars

Les pure players

2010 → Aujourd'hui

Notre terrain de bataille

- **Time To Market**
- **SEO**
- **Webperf, Résilience et disponibilité**
- **Individualisation et contextualisation des pages**
- **Qualité et fraîcheur de la donnée**
- *Disponibilité Produit, Délivrance & Prix*

Un peu d’histoire : retour en 2017

Une centaine de

Devs / PO / Scrum / ...

Des files d’attente interminables pour les MEP

Un gros monolithe vieux de 7 ans

Des ambitions IT inatteignables … Mais des webperf très honorables

Un peu d'histoire : retour en 2017

8 vCPU
32Go Ram

Serveurs d'applications

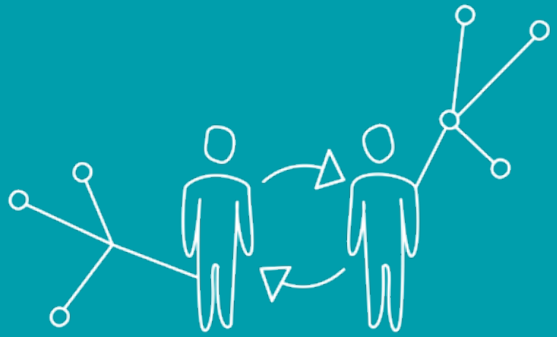
Scaling vertical et horizontal manuel

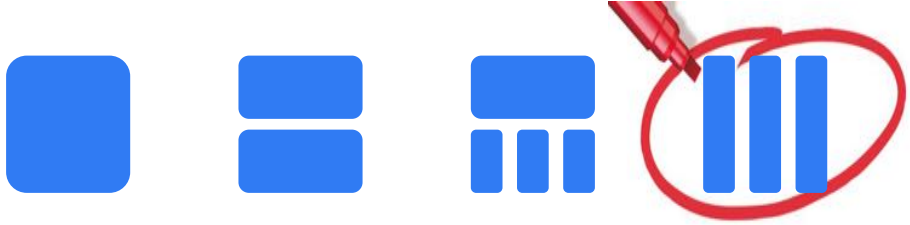
Sticky session

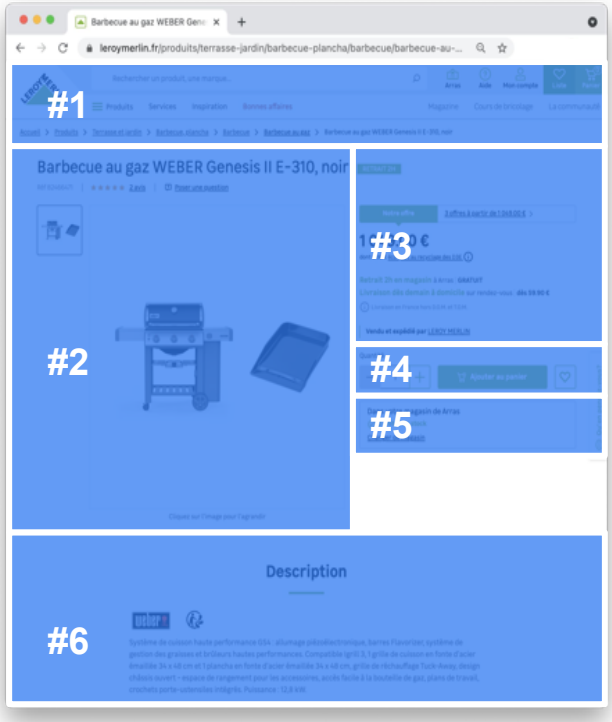
ON REPENSE TOUT ?

NOS CHOIX

L'architecture Micro-Front pour répondre au Time To Market







Fragments list
& associated teams

#1 = Header
Team "Navigation"

#2 = Label & photos
Team "Product"

#3 = Prices
Team "Offer"

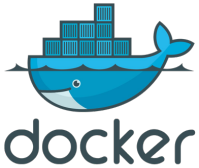
#4 = Add to cart
Team "Cart"

#5 = Availability
Team "Stock"

#6 = Description
Team "Product"

INFRASTRUCTURE

Cloud Computing

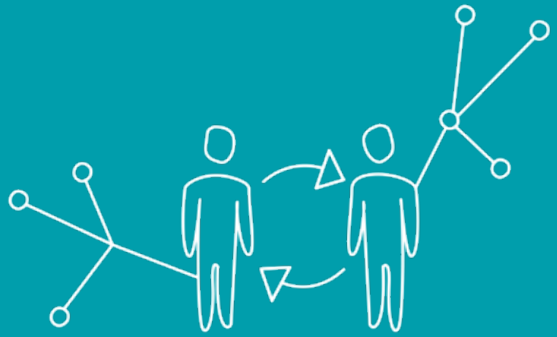


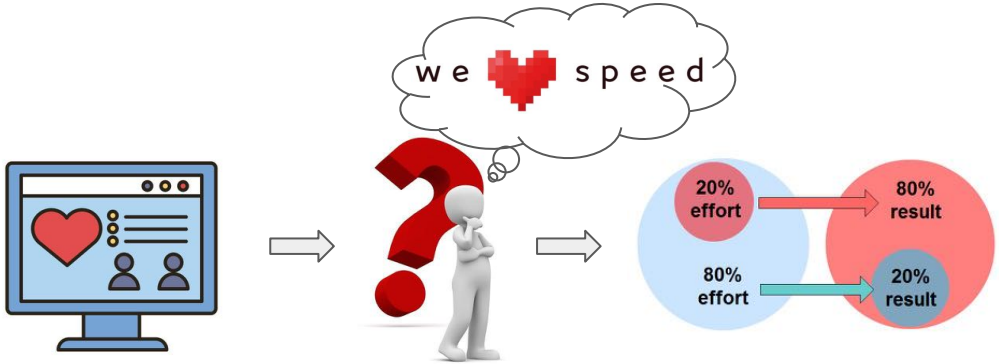
Auto Scaling Horizontal

Stateless

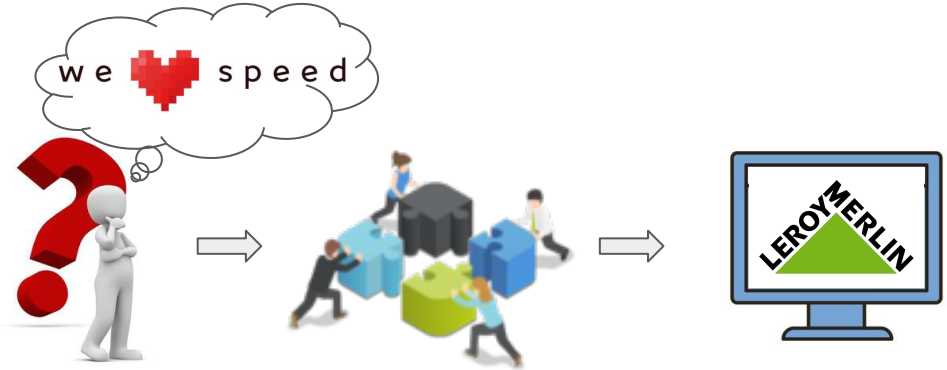
Programmation réactive

DÉMARCHE WEBPERF





La démarche webperf habituelle

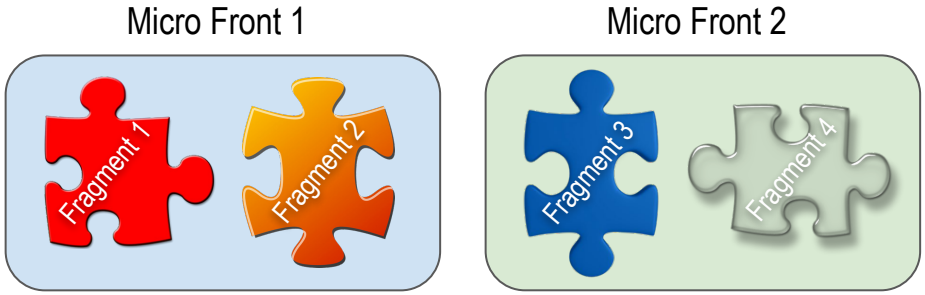


La démarche webperf appliquée lors de notre refonte

VOUS AVEZ DIT MICRO FRONT ?

MICRO FRONT ?

https://www.leroymerlin.fr/we-love-speed/



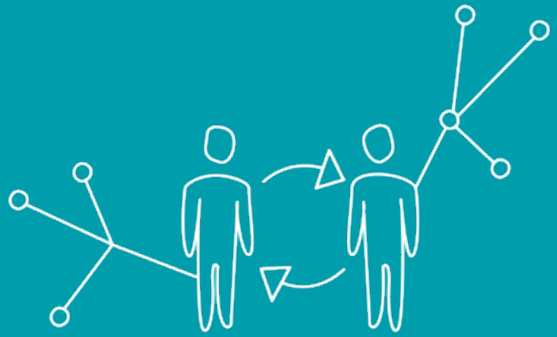
Un Micro Front :

- est une micro application autonome
- délivre un ou plusieurs fragments
- a son propre cycle de vie (dev ► prod + run)
- est déployé dans un container
- est scalable horizontalement

Un fragment :

- est un composant métier autonome
- peut être ajouté/supprimé sans impact sur le reste de la page

PARTIS PRIS



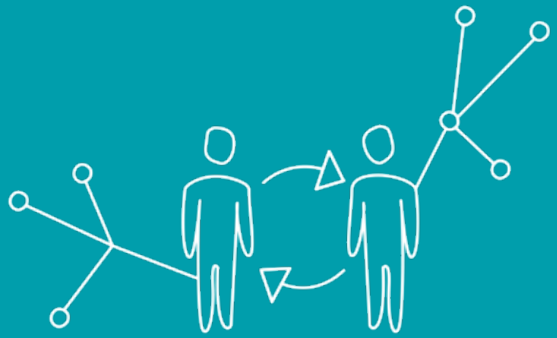
Des fragments autonomes et indépendants

- Des fragments dans différentes technologies.
- Responsables de charger leur données.
- Embarquent leur propre CSS et JS.

Des pages contextualisées et individualisées

- Pas de contenu HTML en cache CDN
- Contenu personnalisé généré côté serveur.
- Contrôle très fin de la fraîcheur de la donnée.

LES PROBLÉMATIQUES



Autonomie et indépendance des fragments :

- Multiplication des ressources statiques
- Duplication de code (css / js)
- Pas de partage de données entre fragments
- Multiplication des appels aux API

Pas de contenu HTML en cache CDN

- Latence supplémentaire pour remonter à nos serveurs
- Charge plus importante sur les serveurs
- Temps de réponse plus important qu'un CDN

QUELLES TECHNO FRONT ?

TECHNOLOGIE FRONT

SEO	Des pages SSR qui se chargent rapidement
Mobile First (55% vs 45%)	Alléger le poids des pages. Limiter les traitements côté client.
Peu de pages vues par visite (~7)	Charger un framework pour afficher 7 pages ?
Developer XP / End User XP	Quel framework pour quelle expérience développeur / client ?
Pérennité techno	Montée de version ? Multi techno ?
Architecture Micro Front	Isolation des fragments vs Framework

TECHNOLOGIE FRONT

- SEO
- Mobile First (55% vs 45%)
- Peu de pages vues par visite (~7)
- Developer XP / End User XP
- Pérennité techno
- Architecture Micro Front

✓ 95% des pages

Home page
Pages produits
Pages éditoriales
Moteur de recherche
...

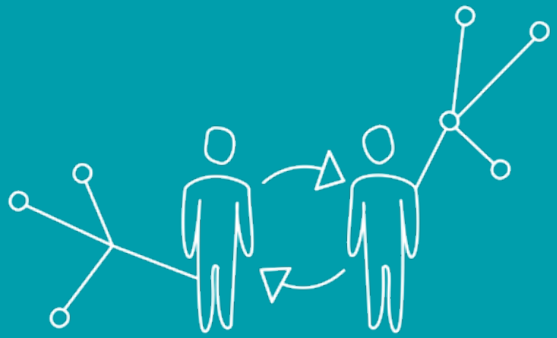
✓ 5% des pages

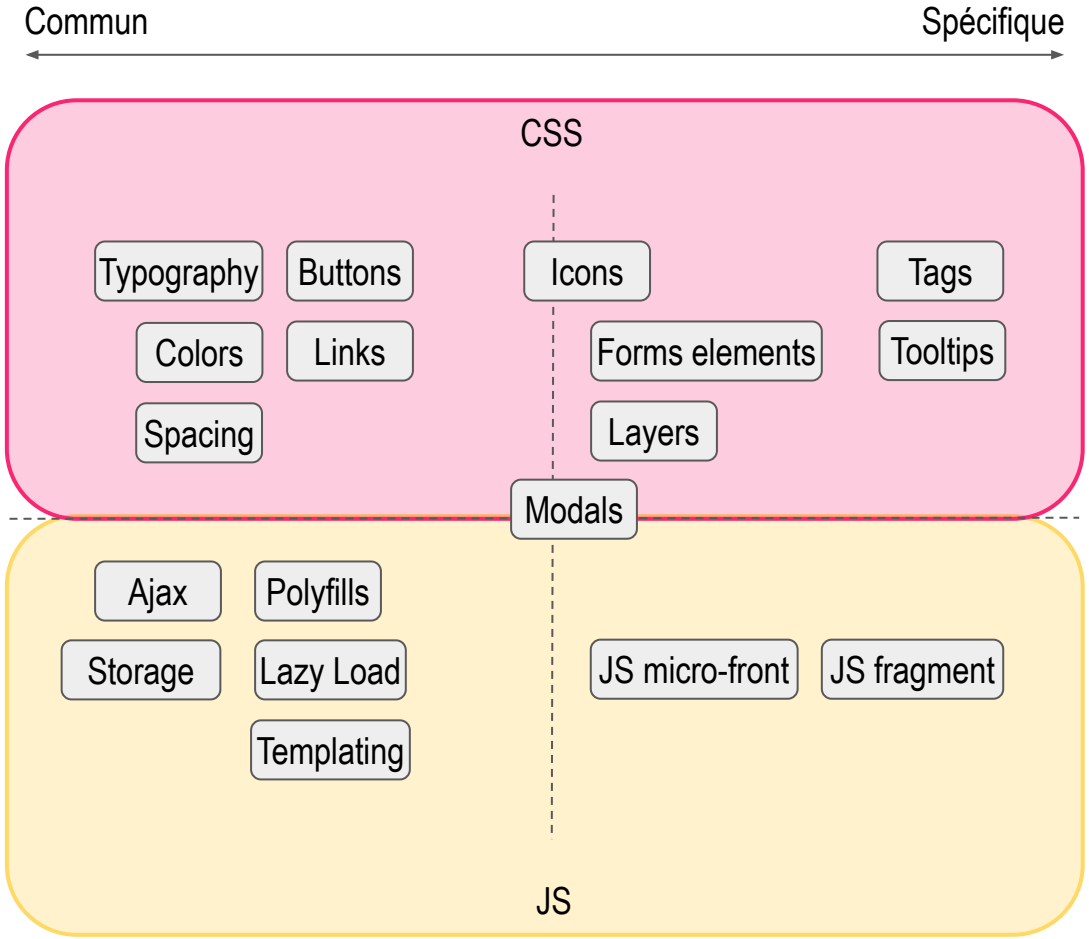
Tunnel de commande
Compte internaute
...

DUPLICATION DE CODE ...
COMMUN OU SPÉCIFIQUE ?

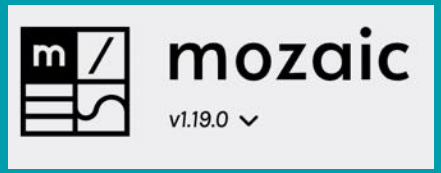
RÉPARTITION DES RESSOURCES

Appel **commun** systématique
OU appel **spécifique** à l'usage





DISTRIBUTION DES RESSOURCES



Design system

Bibliothèques JS

CSS communs
- Fondations (couleurs, spacing, fontes,...)
- Eléments simples communs (boutons, liens,...)

JS communs
- Bus événementiel
- Classes abstraites
- Objets globaux

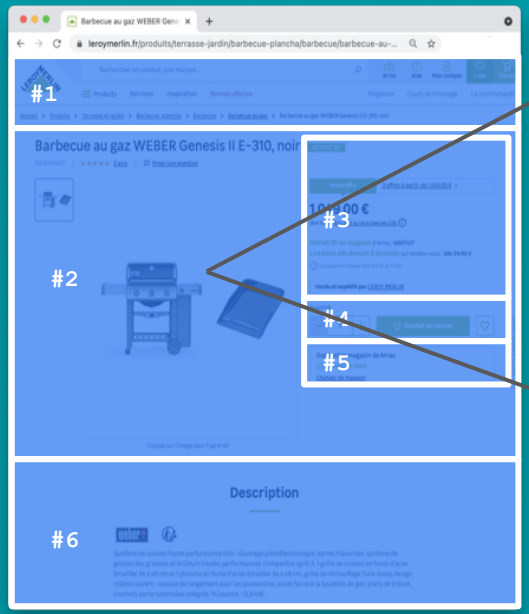
Global Site

CSS spécifiques
- Composants UI spécifiques au contexte (forms, titres,...)

JS spécifiques
- Imports statiques / dynamiques dans le code du fragment

Spécifique Fragment

CONTEXTE FRAGMENTS



Le CSS et le JS de chaque fragment ont leur propre contexte (scoping)

CSS

```css
#component-productdetailmaincdp .kl-swiper__slider {
    display: -webkit-box;
    display: -ms-flexbox;
    display: flex
}

#component-productdetailmaincdp .kl-swiper__slider--center {
    -webkit-box-pack: center;
    -ms-flex-pack: center;
    justify-content: center
}
```

JS

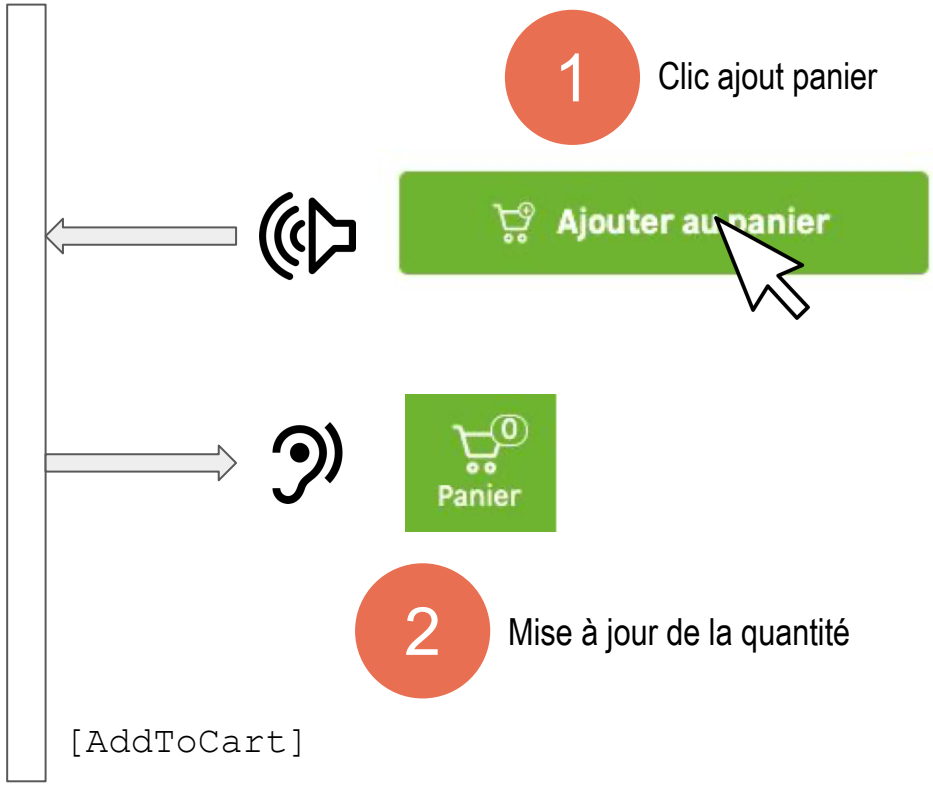
```js
export class ProductDetailMainCdp extends lm.Composant {
    constructor(id) {
        super(id);
        //...
    }
};

lm.DOMReady(function () {
    new ProductDetailMainCdp("productdetailmaincdp");
    initEnergyPopin();
});

export default ProductDetailMainCdp;
```

BUS ÉVÉNEMENTIEL

Permettre aux fragments de communiquer entre eux sans adhérence forte



PAS DE PERFORMANCE
SANS RESILIENCE

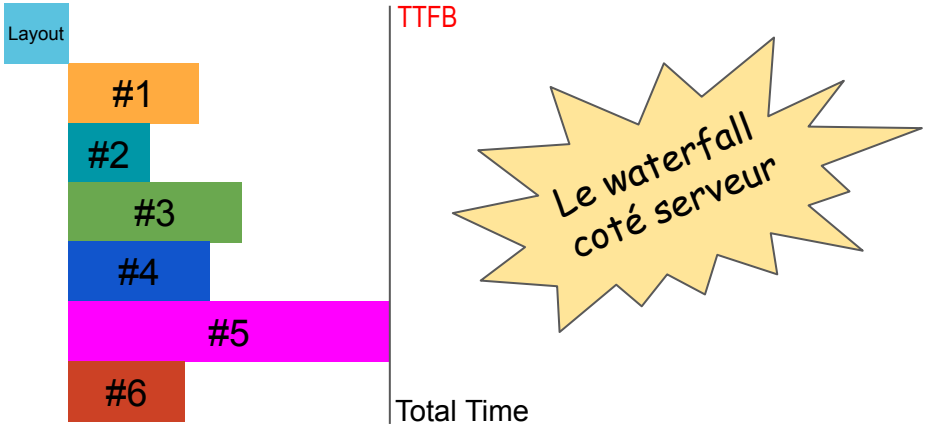
PERFORMANCE

- Appels séquentiels

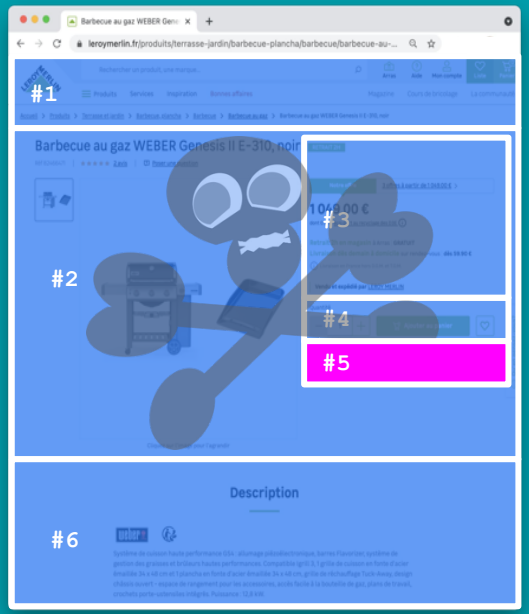
WATERFALL SERVER

- Appels séquentiels

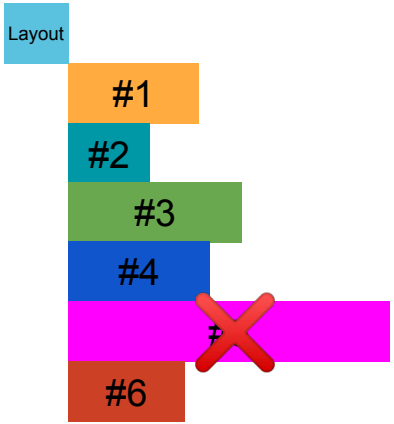
- Parallélisation des appels



RESILIENCE

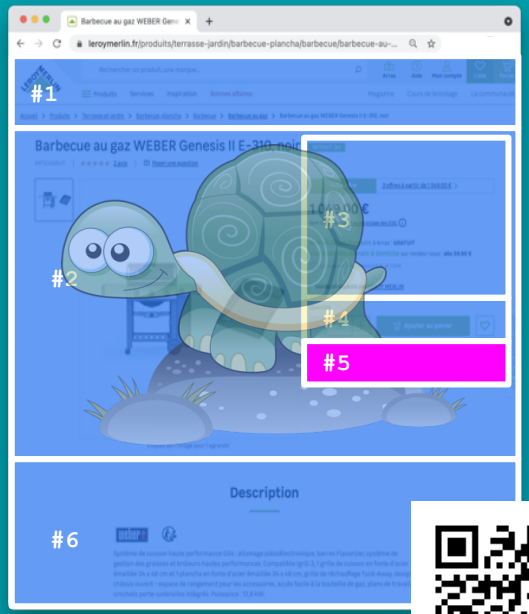


- Hypothèse : chaque fragment a un taux de dispo de 99.99%

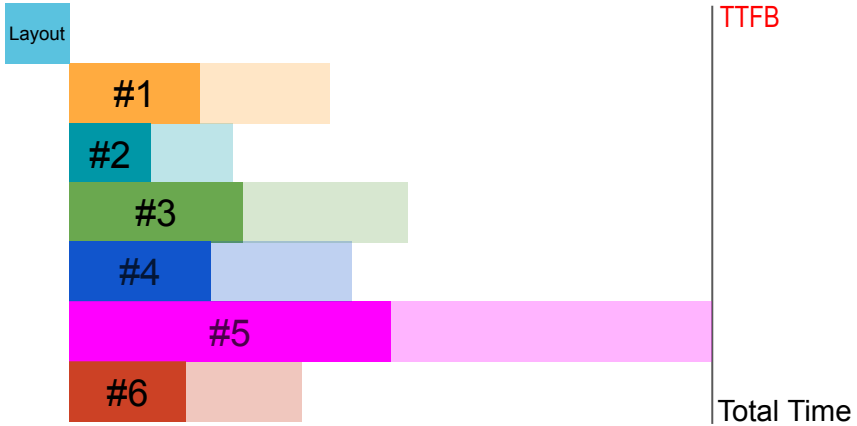


Avec 6 fragments, ma disponibilité descend à **99.94%** (99.99% ^ 6)
Avec 25 fragments, ma disponibilité descend à 99.75% (99.99% ^ 25)

PERFORMANCE



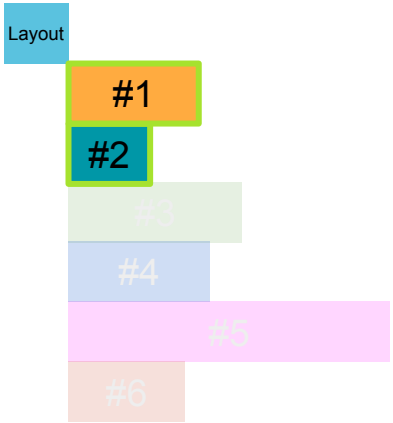
- Hypothèse : chaque fragment a un temps de réponse qui va du simple au double



Mon temps de réponse total peut être doublé

RESILIENCE

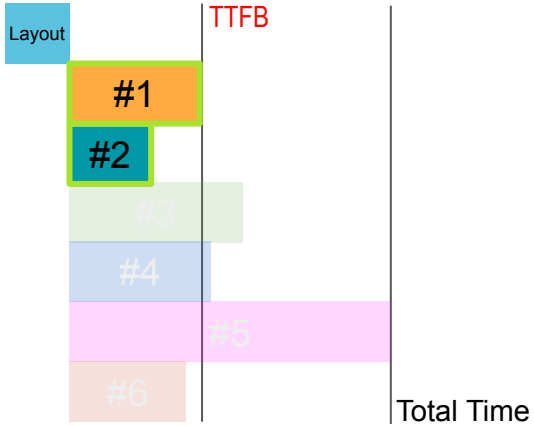
- Est-ce que mes fragments ont tous la même importance ?



Avec 6 fragments dont 2 primaires, ma disponibilité remonte à **99.98%**
Avec 25 fragments dont 3 primaires, ma disponibilité remonte à 99.97%

PERFORMANCE

- Est-ce que mes fragments ont tous la même importance ?

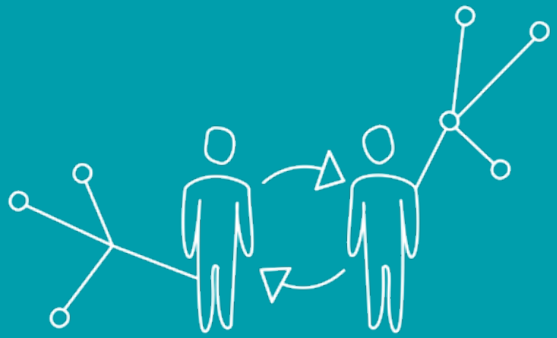


Avec 6 fragments dont 2 primaires, ma disponibilité remonte à **99.98%**
Avec 25 fragments dont 3 primaires, ma disponibilité remonte à 99.97%

Le TTFB est uniquement impacté par le temps de réponse des fragments primaires
Le téléchargement des ressources critiques est précipité

CACHE OU PAS CACHE ?

CACHE OU PAS CACHE



Le cache immutable c'est génial

- Modification d'une ressource statique = changement d'url

Le cache avec TTL, c'est le début des ennuis

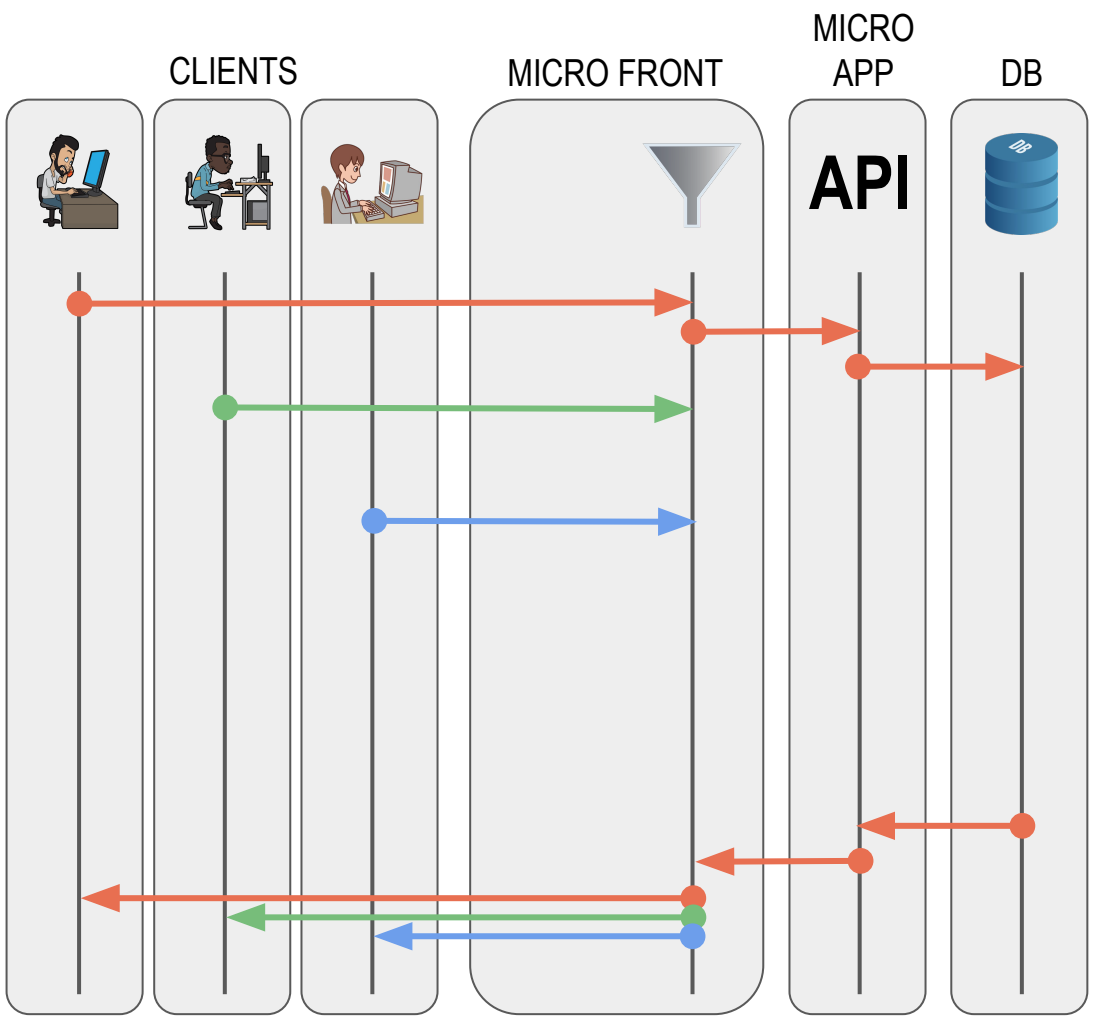
- Problème de fraîcheur de la donnée.
- Déphasages de données sur les architectures microservices
- Complexité de l'architecture avec un cache distribué

Mais ...

+ Amélioration de la perf
+ Limitation de la charge sur l'infrastructure

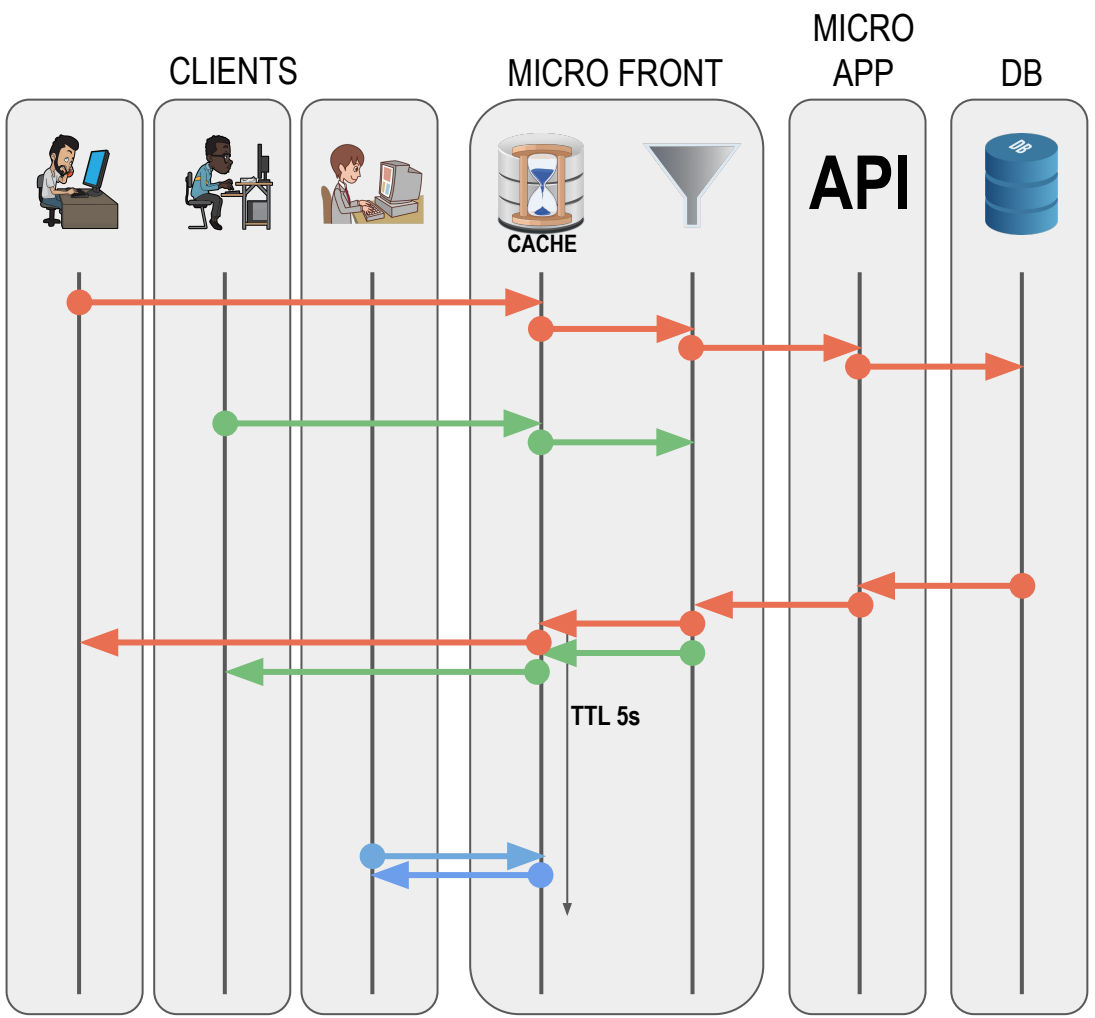
API

- Éviter les appels simultanés aux mêmes ressources



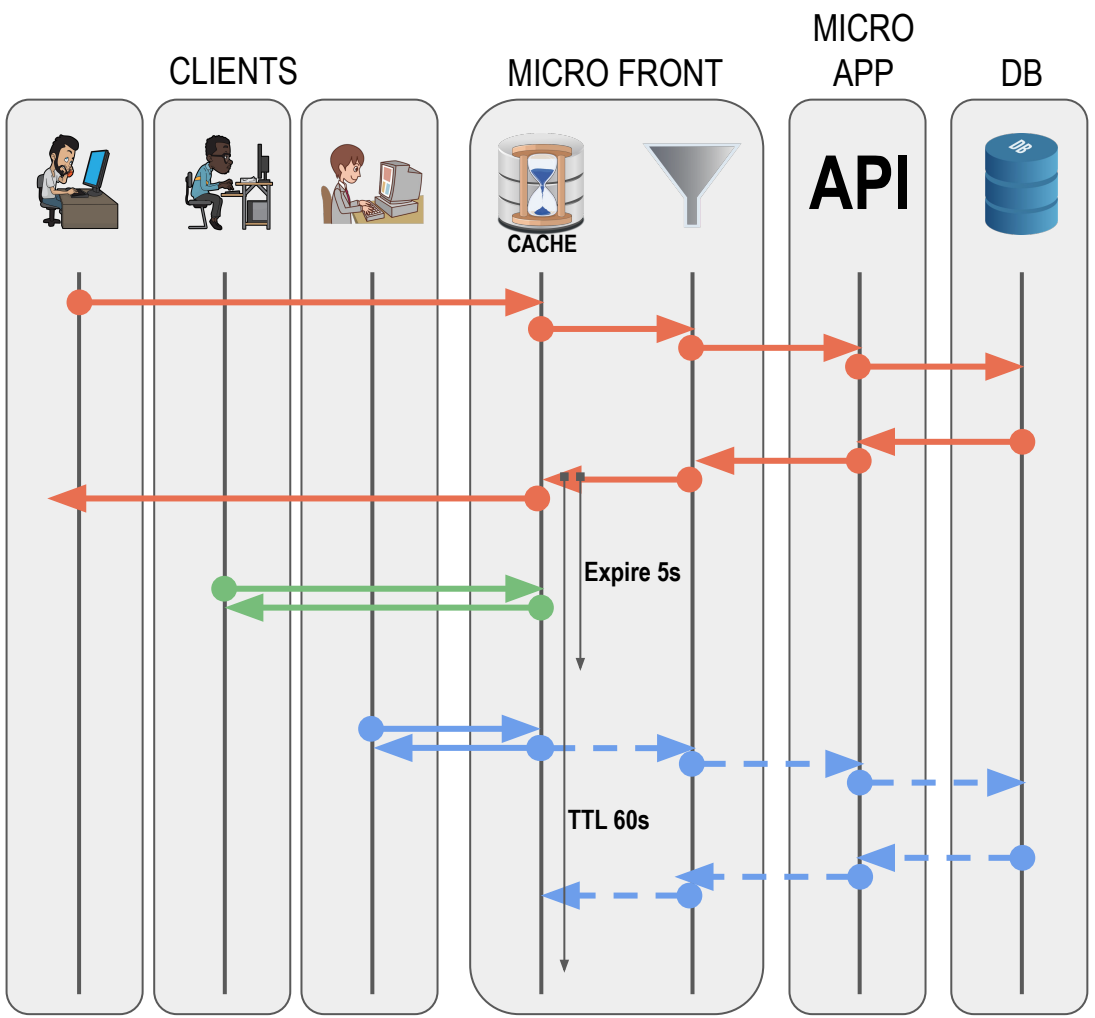
API + CACHE

- Éviter les appels simultanés aux mêmes ressources
- Réduire les appels aux API
- Diminuer les temps de réponse



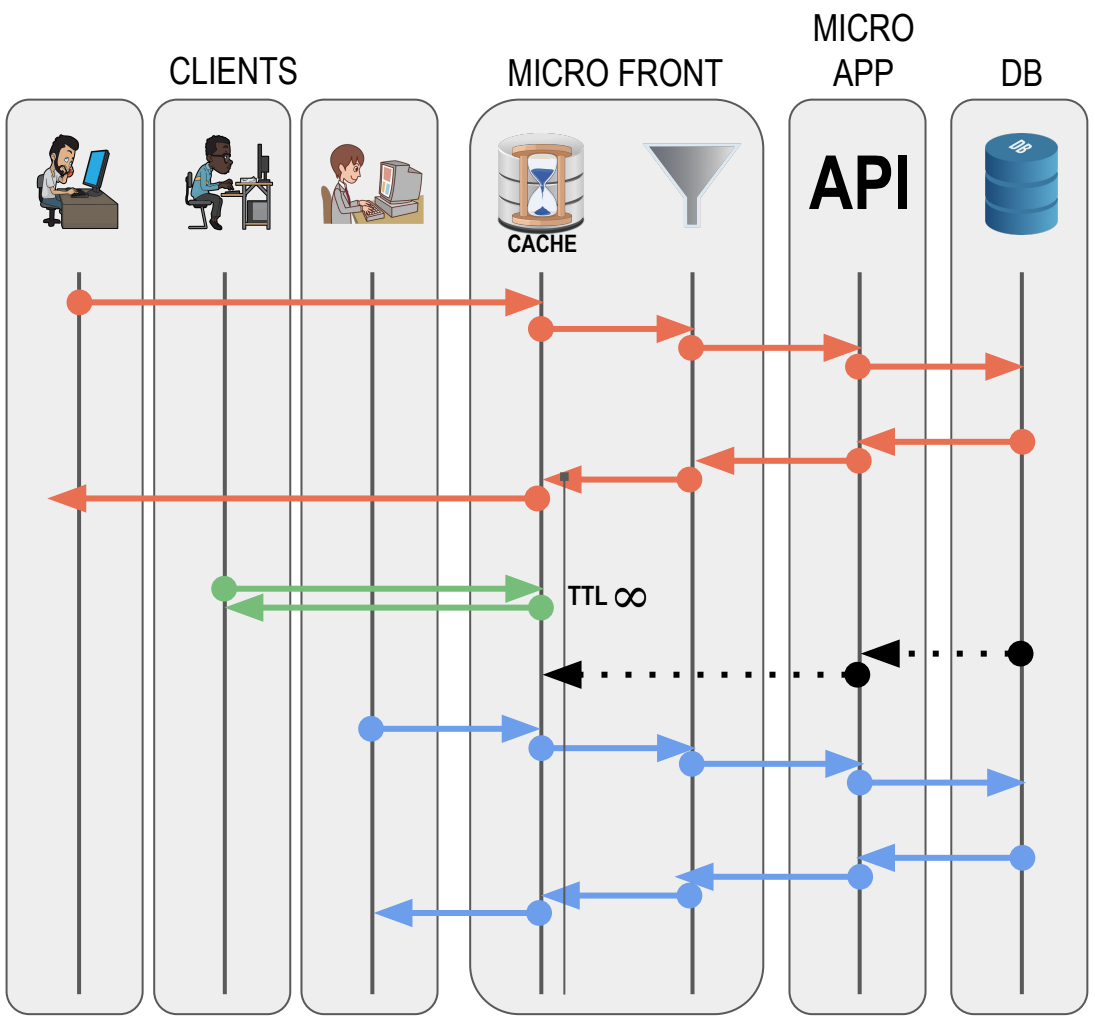
API + CACHE + ASYNC

- Éviter les appels simultanés aux mêmes ressources
- Réduire les appels aux API
- Diminuer les temps de réponse
- Éviter toute latence lors des évictions de cache



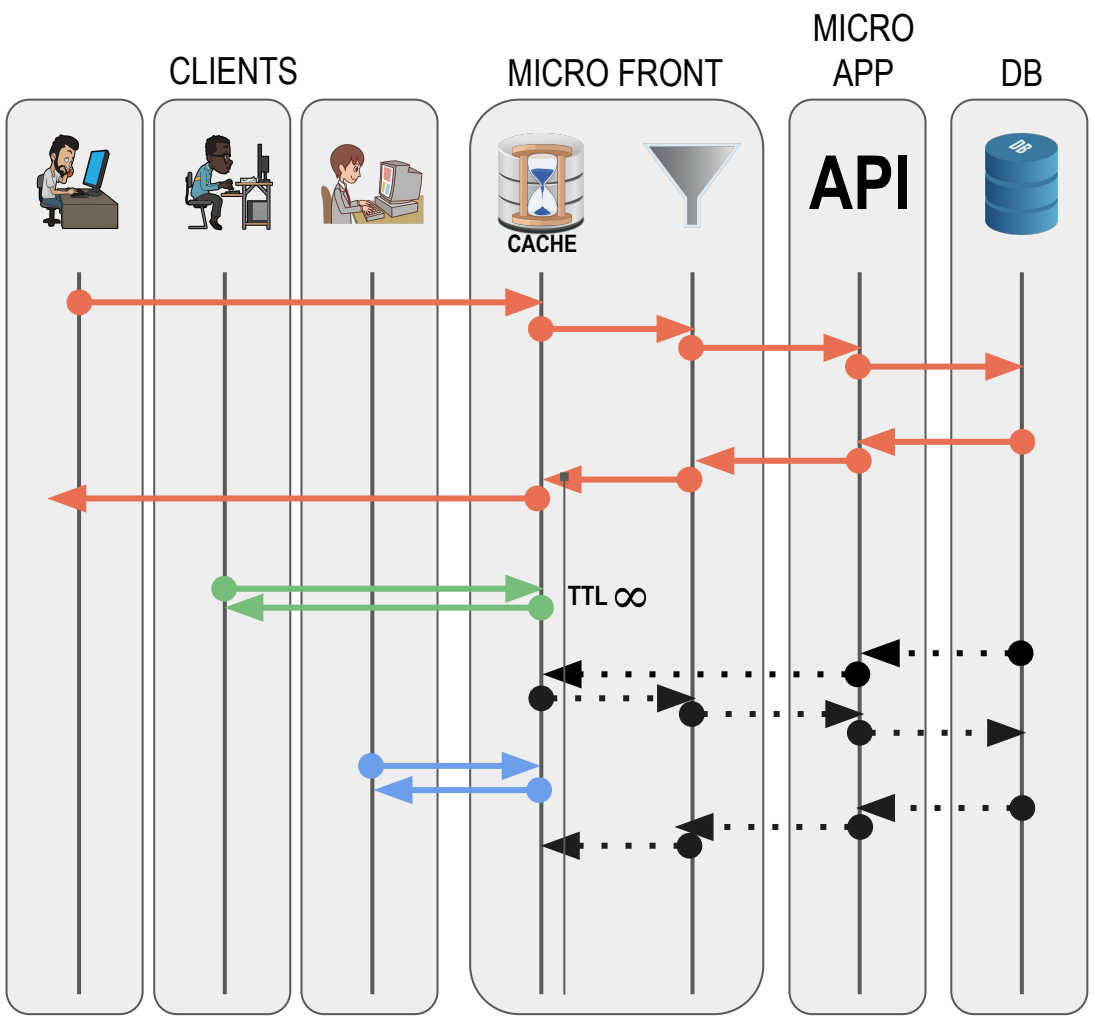
EVENT DRIVEN CACHE

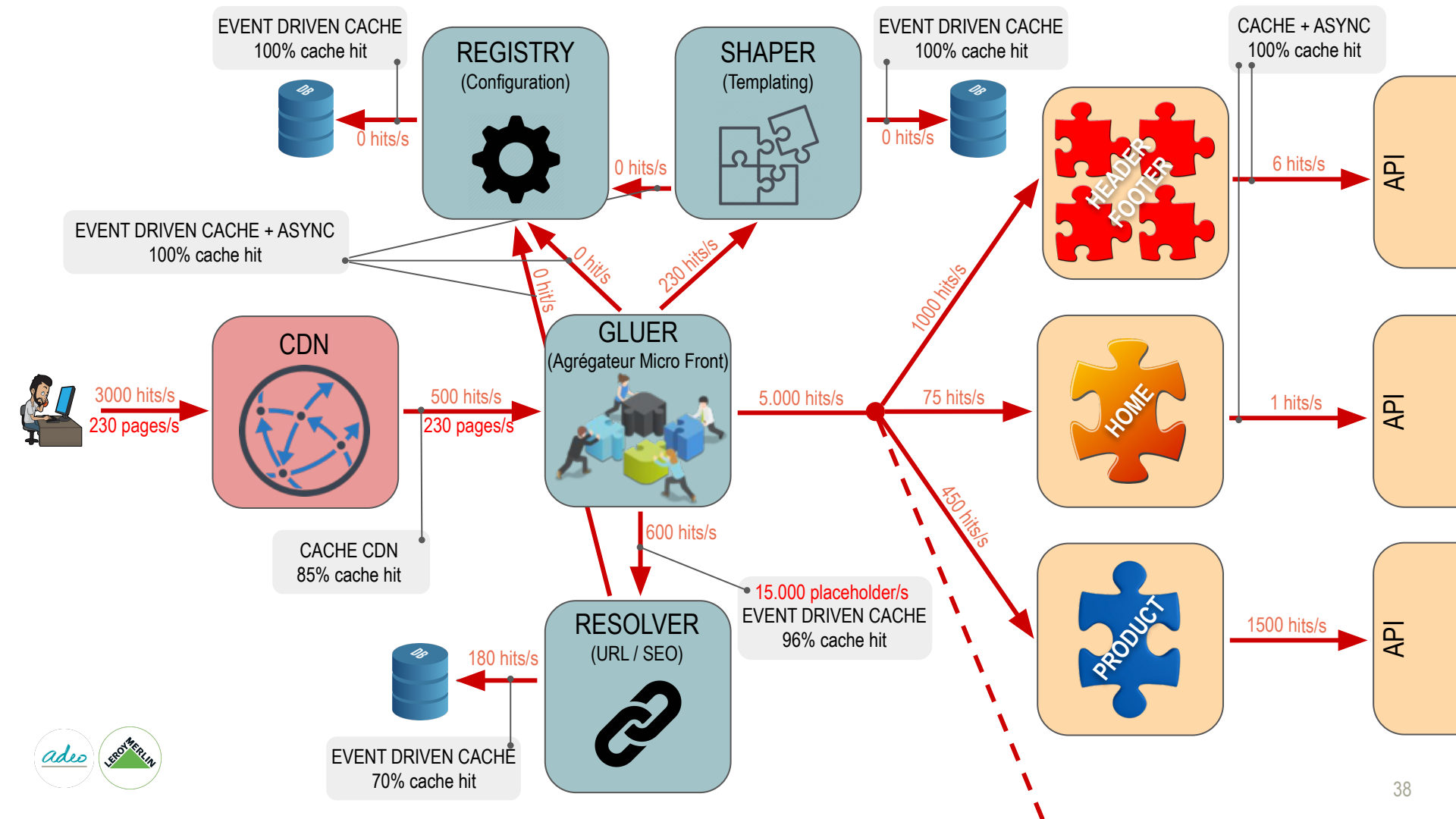
- Éviter les appels simultanés aux mêmes ressources
- Réduire les appels aux API
- Diminuer les temps de réponse
- Avoir une donnée toujours à jour



EVENT DRIVEN CACHE + ASYNC

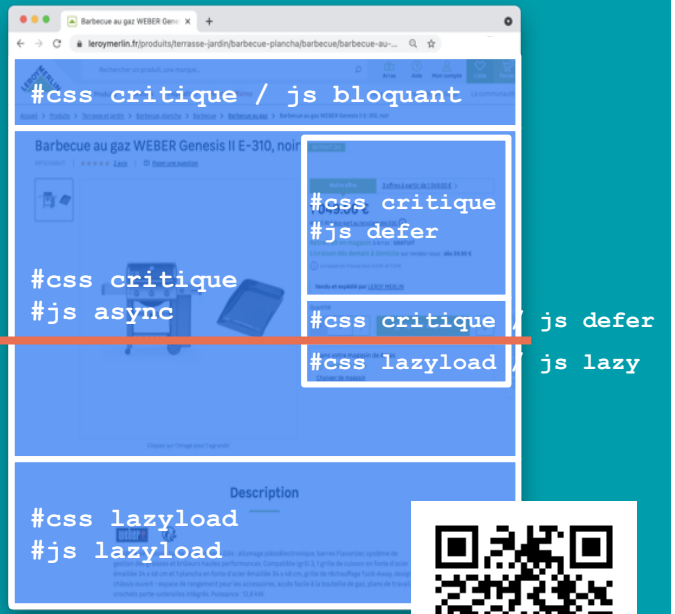
- Éviter les appels simultanés aux mêmes ressources
- Réduire les appels aux API
- Diminuer les temps de réponse
- Avoir une donnée toujours à jour
- Éviter toute latence lors des évictions de cache




EVENT DRIVEN CACHE
100% cache hit
DB
0 hits/s
REGISTRY
(Configuration)
SHAPER
(Templating)
EVENT DRIVEN CACHE
100% cache hit
DB
0 hits/s
CACHE + ASYNC
100% cache hit
0 hits/s
EVENT DRIVEN CACHE + ASYNC
100% cache hit
0 hit/s
0 hit/s
230 hits/s
HEADER
FOOTER
6 hits/s
API
CDN
GLUER
(Agrégateur Micro Front)
1000 hits/s
3000 hits/s
230 pages/s
500 hits/s
230 pages/s
5.000 hits/s
75 hits/s
HOME
1 hits/s
API
CACHE CDN
85% cache hit
600 hits/s
450 hits/s
15.000 placeholder/s
EVENT DRIVEN CACHE
96% cache hit
RESOLVER
(URL / SEO)
PRODUCT
1500 hits/s
API
DB
180 hits/s
EVENT DRIVEN CACHE
70% cache hit

GESTION DES RESSOURCES STATIQUES

RESSOURCES CRITIQUES



Trois niveaux de criticité gérés via la configuration de fragment

- Nécessaire à l'affichage au dessus de la ligne de flottaison : **Critique / Server push**

```
<link href="common.css" rel="stylesheet" type="text/css">
```

```
<link href="fragment1.css" rel="stylesheet" type="text/css">
```

```
<script src="common.js"></script>
```

- Scripts au dessus de la ligne de flottaison **async / defer**

```
<script src="fragment2.js" async defer></script>
```

- CSS / Scripts en dessous de la ligne de flottaison **lazy load**

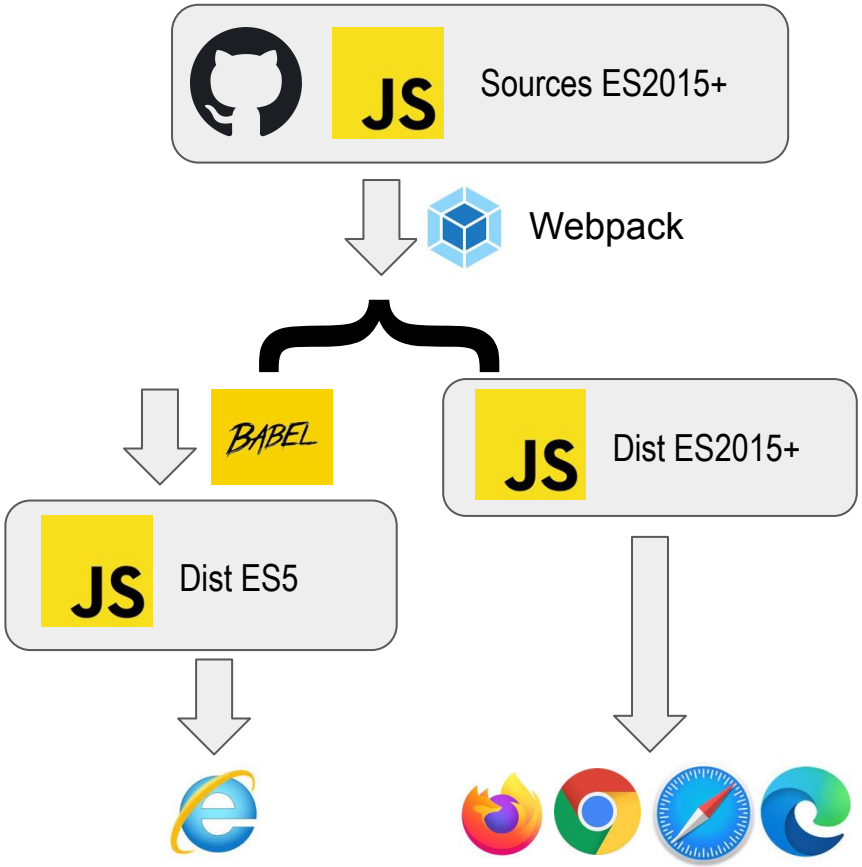
```
<script data-src="fragment3.js" class="lazy-src"></script>
```

OPTIMISATION DU JS PAR NAVIGATEUR

Identification du navigateur côté serveur par rapport à son User Agent

30% de poids de fichier gagné en moyenne pour les navigateurs récents

ES5 pour IE11 et consorts, ES2015+ pour les autres



LES IMAGES ET LE CDN

fastly

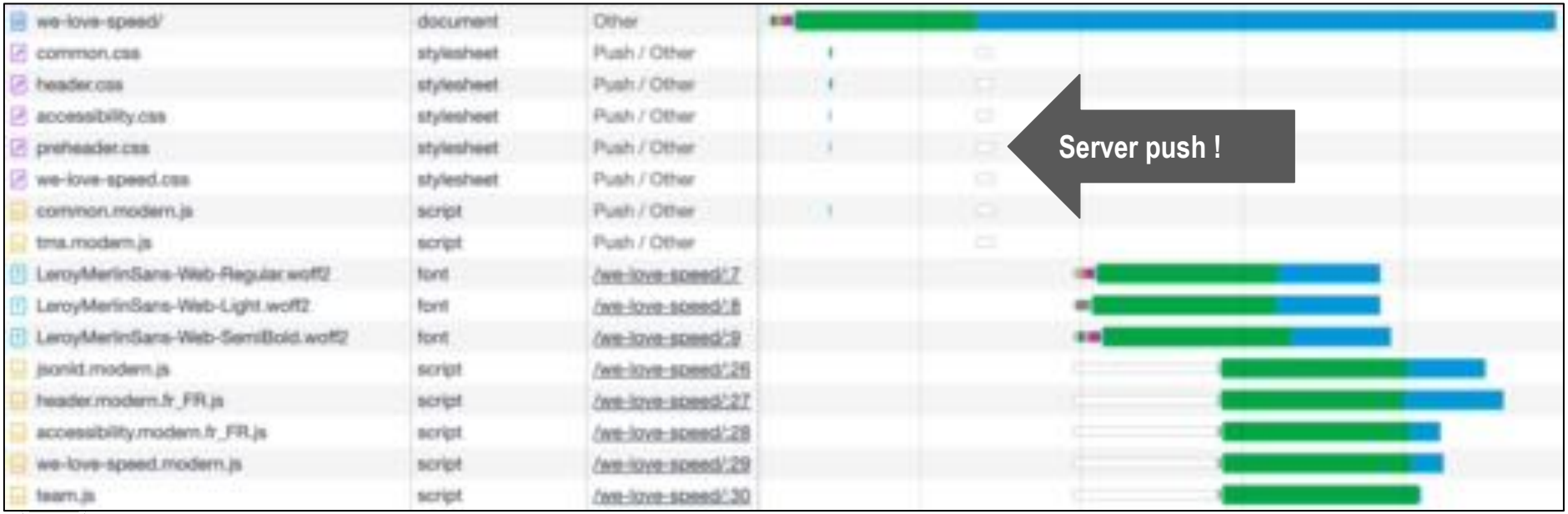
Images responsives

- Redimensionnement et cropping des images
- Compression des images
- Conversion au format WEBP si le navigateur est compatible

```
<picture class="col-12 col-m-12 col-l-8 col-xl-8 col-xxl-8 o-maxarea__hero">
   <source srcset="https://media.adeo.com/media/1647088.jpeg?width=416&crop=16:9,smart" media="(max-width: 414px)">
   <source srcset="https://media.adeo.com/media/1647088.jpeg?width=768&crop=16:9,smart" media="(max-width: 1023px)">
   <source srcset="https://media.adeo.com/media/1647088.jpeg?width=620&crop=8:5,smart"  media="(max-width: 1280px)">
   <source srcset="https://media.adeo.com/media/1647088.jpeg?width=811&crop=8:5,smart"  media="(max-width: 1920px)">
   <img src="https://media.adeo.com/media/1647088.jpeg?width=811&crop=8:5,smart" class="o-maxarea__img">
</picture>
```

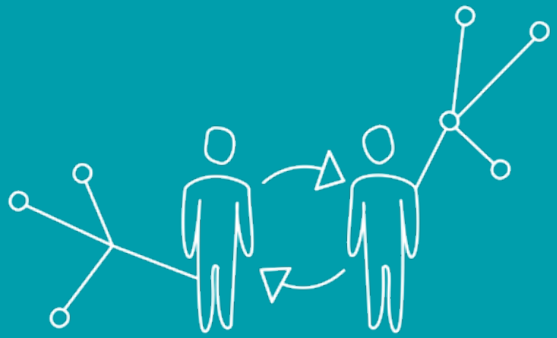
SERVER PUSH

- Déprécié ? Non selon Google en attendant mieux (Early hints)
- Ne faire du server push que sur les ressources critiques sinon risque de détérioration de la webperf



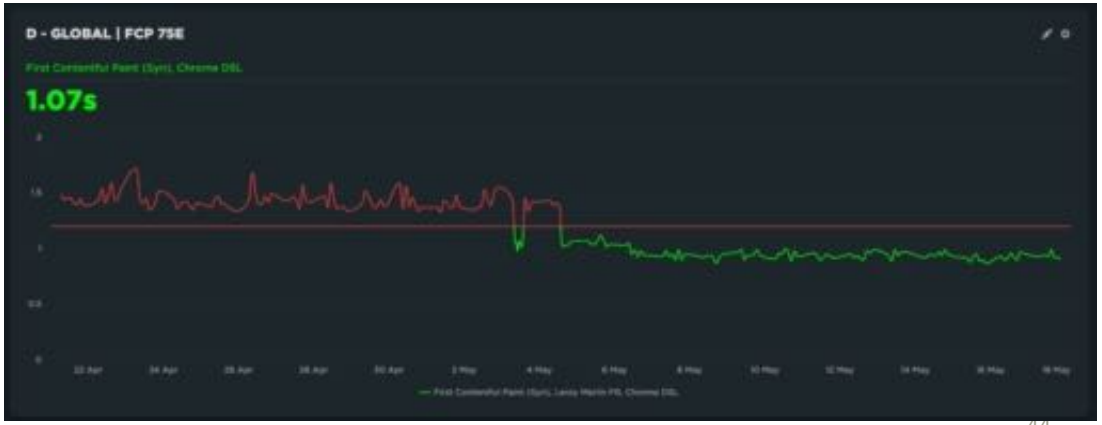
SERVER PUSH

- Diminution du DOM CONTENT LOADED
- Diminution du FIRST CONTENTFUL PAINT



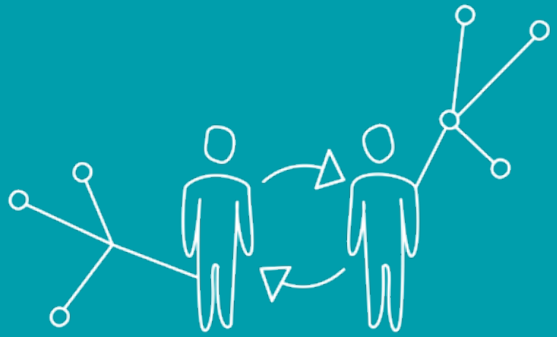
DOM CONTENT LOADED

FIRST CONTENTFUL PAINT



SERVER PUSH

- Diminution du DOM CONTENT LOADED
- Diminution du FIRST CONTENTFULL PAINT
- Augmentation du CUMULATIVE LAYOUT SHIFT



CUMULATIVE LAYOUT SHIFT

AUTONOMIE DES FRAGMENTS & GESTION DES TIERS

TAG MANAGEMENT

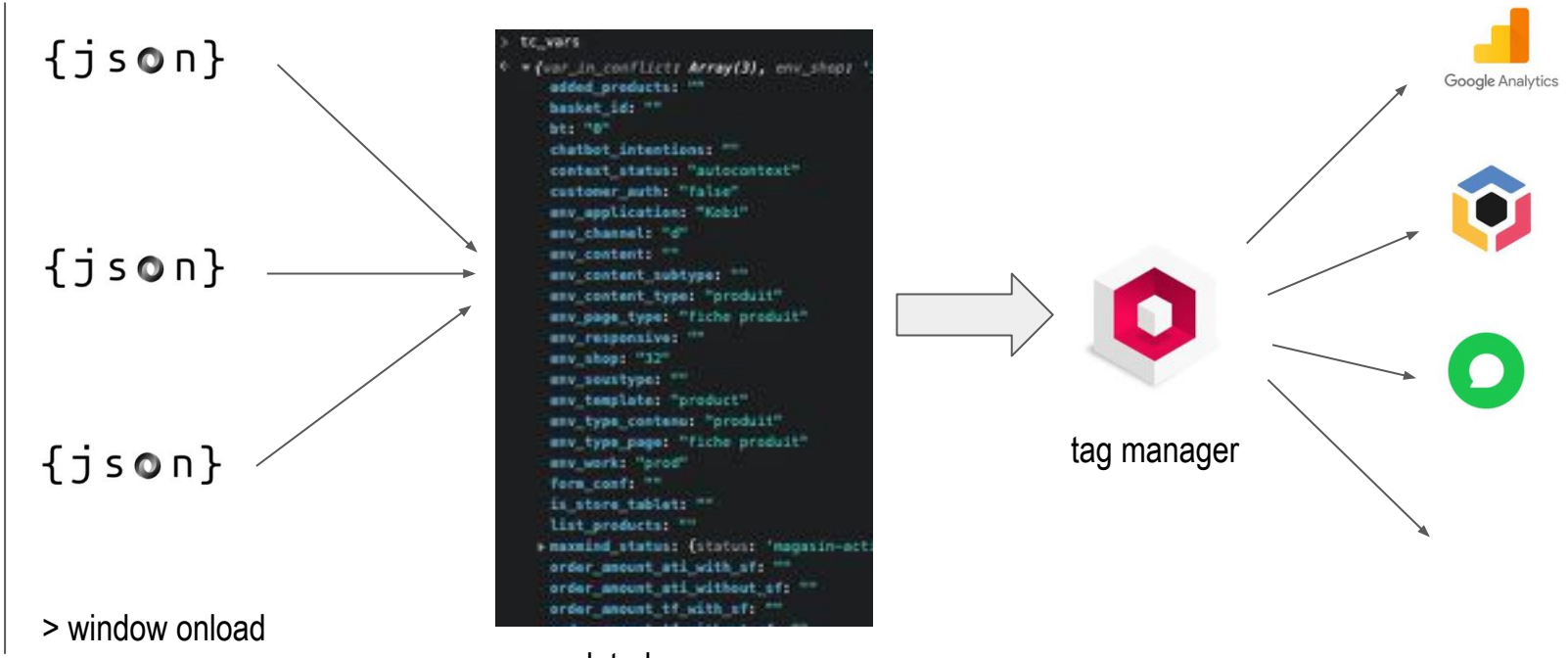
60% du poids des pages de leroymerlin.fr provient de ressources tierces

- Cartographie
- Webanalyse / Comportemental
- Tests AB
- Publicité / présence externe
- Outils externes (chatbot,...)
- DMP
- ...

→ **MONOLITHE !**

TAG MANAGEMENT

Agrégation des JSONs pour construire le data-layer et activation du Tag Manager

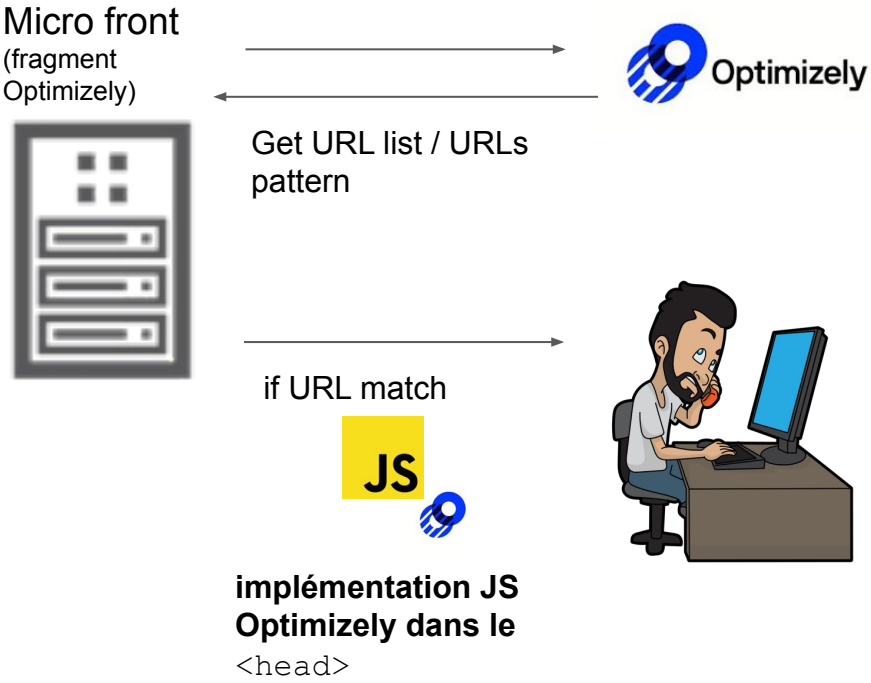


AB TEST

Un AB test n'impacte jamais 100% des pages

Pourtant l'implémentation classique de ce type d'outils revient à un script bloquant l'affichage sur toutes les pages

Solution : utiliser l'API de l'outil pour connaître les URLs impactées par les tests



AUTONOMIE DES ÉQUIPES & ORGANISATION

CULTURE COMMUNE ?

Une vision différente par métier...

La perf serveur est bonne

Qualité

Le load time dans GA n'est pas énorme

Analytics

C'est quoi les KPIs à surveiller ?

PO / Scrum

Nous avons tous des iPhones et des Macs

UX / UI

Je regarde CrUX

SEO

Est-ce que mon site est rapide ?

Sponsors

J'utilise Lighthouse

Développeur

Tout le monde a la fibre

Métier

4 PILIERS D'UNIFORMISATION

1

Connaître ses utilisateurs

2

Analyser son site

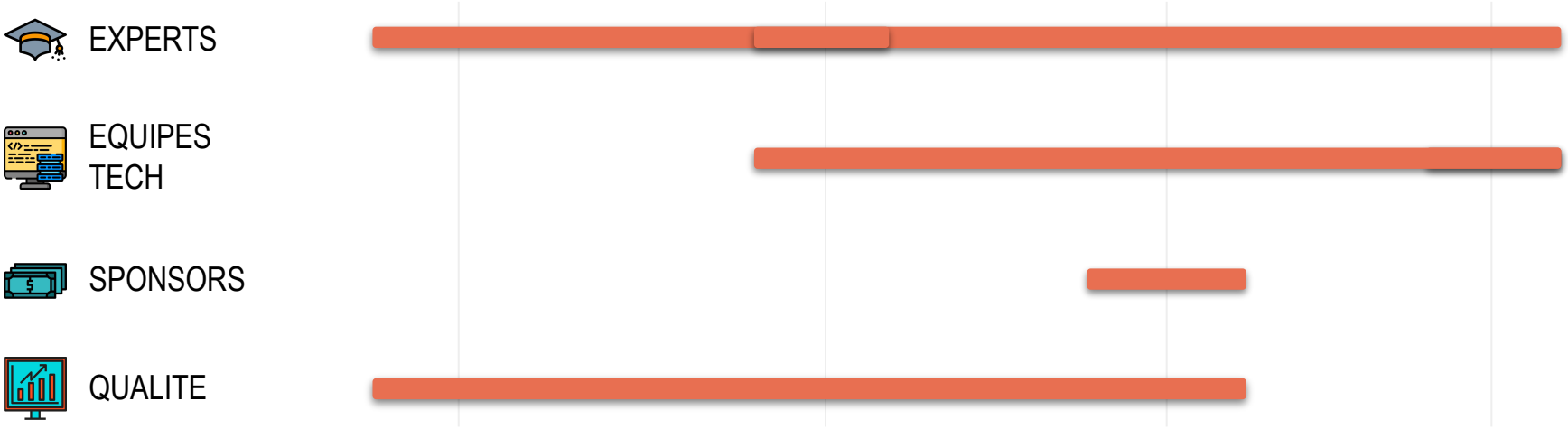
3

Communiquer ses indicateurs

4

Animer / challenger

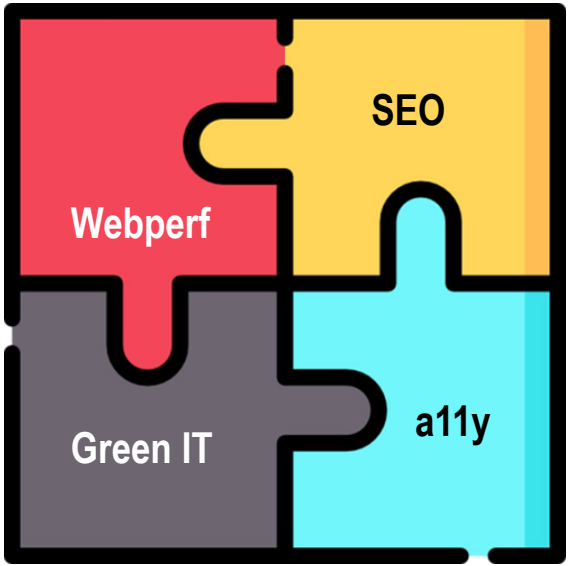
DISTRIBUTION DES TACHES



LE RSE POUR UNE ADHÉSION PLUS FORTE

Leroy Merlin s'engage sur son impact environnemental et sociétal

- Priorisation du Green IT et de l'accessibilité en 2022
- Sensibilisation / formation de tous les métiers



Et donc, l’architecture micro-front, un levier pour la webperf ?

- Meilleur gestion des ressources critiques
- Meilleur contrôle du rendu des pages
- Meilleur Time To First Byte
- Meilleure résilience

Mais préparez-vous à un changement global de votre organisation !

Merci pour votre attention !
QUESTIONS ?