

Guide Pratique

Design for Speed ⚡💻📱

Pourquoi et comment concevoir des sites web plus rapides ?

USERADGENTS



20 ans de mobile 20 ans de mutations

Du wap à la 5G, du SMS Marketing aux notifications, des sites dédiés aux apps, des MMS aux stories, au streaming, au live, à la réalité augmentée... en 20 ans, le web mobile **a considérablement évolué.**

Un monde mobile first voire mobile only

En 2015, les requêtes sur mobile dépassent celles effectuées sur ordinateur fixe.

57% utilisent le + souvent leur smartphone pour accéder au web.

Des réseaux plus rapides, des smartphones toujours plus performants, le mobile s'est imposé comme le **premier écran de consultation du web.**

Aujourd'hui, c'est **plus de la moitié du trafic web qui provient du mobile.**

La vitesse, un enjeu majeur

A l'heure du mobile first, voire mobile only, où le **streaming** devient la norme, où les **contenus** deviennent de **plus en plus lourds** pour s'adapter à des écrans aux **résolutions** toujours plus grandes, la **puissance exponentielle des réseaux** et le **développement de data centers** aux quatre coins du monde **ne suffisent pas** à répondre à l'enjeu majeur de vitesse !



Le temps de chargement d'un site a globalement **augmenté de 21%** ces dernières années !



50%

Aujourd'hui, les images représentent + de 50% du poids total des pages web.

x24

De la traditionnelle résolution 480p au UHD 4K, le nombre de pixels qui composent l'image a été multiplié par 24 !





58%

Le streaming vidéo pèse pour 58%
de la bande passante mondiale !

15%

est consommé par Netflix !

Des consommateurs impatients & intransigeants

qui attendent de leur expérience mobile qu'elle soit :

+ Instantanée **+ Intuitive** **+ Personnalisée** **+ Immersive**

Un accès ultra
rapide au contenu

de manière fluide
& agréable

un contenu adapté
à l'utilisateur

tirant profit des fonctionnalités
natives du smartphone (caméra,
synthèse vocale...)

Pas de pitié pour les moins rapides

En France, le **temps de chargement moyen des sites mobiles est de 8,8 secondes** en connexion 3G. Lorsqu'on sait que **Google recommande 1 à 3 secondes** et qu'une étude Google analyse que 53% des visites sont abandonnées si la page excède ce temps de chargement, il est **primordial d'optimiser son site** pour atteindre cet objectif de vitesse.

53% des visites
sont abandonnées si la
page met + de 3 sec
à charger



67%

abandonnent leur panier à cause de la
lenteur du site

7%

Pour chaque seconde, le taux de
conversion chute de **7%**, le nombre de
pages vues de **11%** & la **satisfaction client**
de **16%** en moyenne



Amazon

Quand son temps de chargement augmente de 100 ms, soit **0.1 seconde**, c'est une **perte de 1% des ventes**, ce qui aurait représenté en 2016 une perte de **1.360 milliards d'euros**.

Etam

L'enseigne de lingerie communiquait en 2015 un gain de **20% de son taux de conversion** suite à une baisse du temps moyen de chargement de ses pages de 0.7 secondes.

La vitesse a aussi un impact sur le SEO & le SEA

Depuis Juillet 2018, le **Speed Update** sanctionne les sites trop lents sur mobile.

Depuis Août 2018, un **score de vitesse** sur mobile est présent dans Google Ads.

La conversion et satisfaction du client ne sont donc pas les seules raisons d'optimiser sa vitesse, il s'agit également d'être mieux positionné dans les requêtes.



Alors comment optimiser la **vitesse** et la **performance** de son site web ?

👉 Suivez le guide

Comprendre

Pré-requis : les bases du web

Faisons le point sur les éléments qui composent un site web pour mieux comprendre les enjeux de vitesse.

P.13

Analyser

Audit de performance

Première étape concrète : analyser les performances de votre site via des outils méthodiques.

P.21

Embarquer

Démarche collaborative

Suite à l'audit, fixez vous des objectifs clairs. Pour être dans une démarche speed-first, toutes les équipes doivent être sensibilisées et impliquées.

P.29

Optimiser

Best practices

Suivez nos conseils techniques, UX & UI pour optimiser les aspects cruciaux de votre site web pour une meilleure performance.

P.32

Les solutions du marché

Focus sur les solutions qui s'offrent à vous : site dédié, responsive, dynamic serving, AMP, PWA, PWAMP...

01

Comprendre

Pré-requis : les bases du web

Avant tout, on fait le point sur les bases du web

Pour mieux comprendre les axes d'optimisation que nous vous proposons dans ce guide, retour sur ce qui constitue une page web, comment fonctionnent les requêtes, les ressources des navigateurs & des serveurs et l'hébergement.

**Vous êtes déjà calé.e en la matière ?
Allez direct à la page 21.**

Une page web, c'est quoi ?



+ HTML

(HyperText Markup Language)

Les fichiers html permettent de définir la structure de votre page.

Ex: si vous souhaitez indiquer que votre page se compose d'une image, d'un titre et d'un paragraphe, c'est du HTML.



+ CSS

(Cascading style sheet)

Les fichiers CSS sont utilisés pour déterminer comment s'affichent les éléments de la page que vous avez déclaré dans le HTML.

Ex: si vous souhaitez modifier la couleur d'arrière-plan de vos pages ou les polices utilisées, c'est du CSS.



+ JS

(Javascript)

Les fichiers javascript vont permettre à votre site d'être « dynamique ».

Ex: si vous souhaitez intégrer un formulaire avec auto-complétion, ce sera probablement du JS.

Une page web, c'est aussi des fichiers



MP4 | OGG | WEBM...

+ Vidéos



PNG | JPEG | TIFF | SVG...

+ Images



OTF | TTF

+ Polices

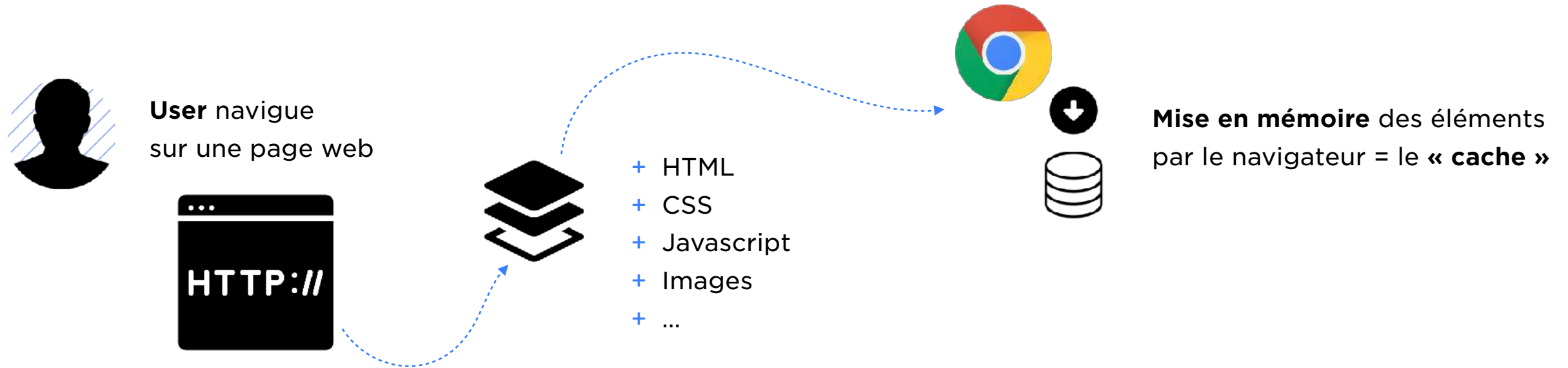


JSON | XML

**+ Échange
de données
textuelles**

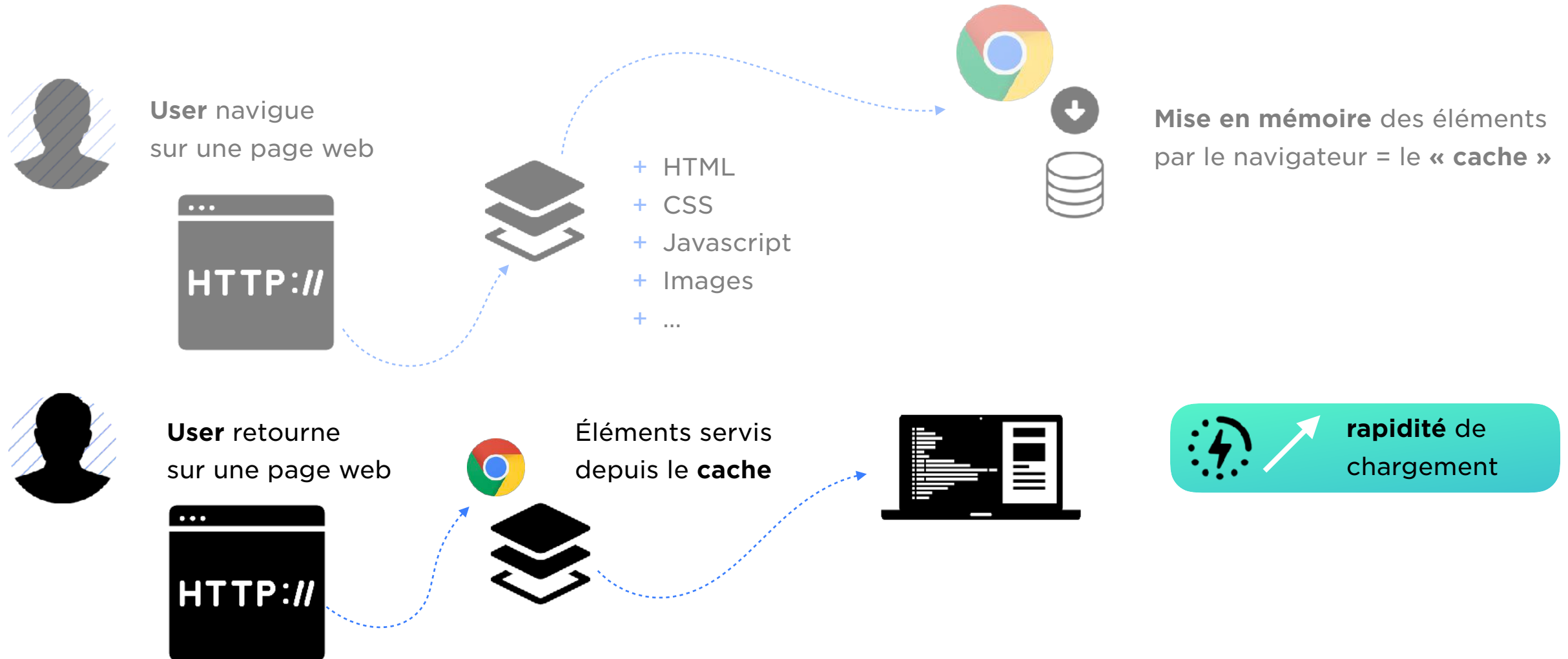
La notion de « Cache » sur un navigateur

Essentielle pour obtenir des performances optimales sur son site web & mobile



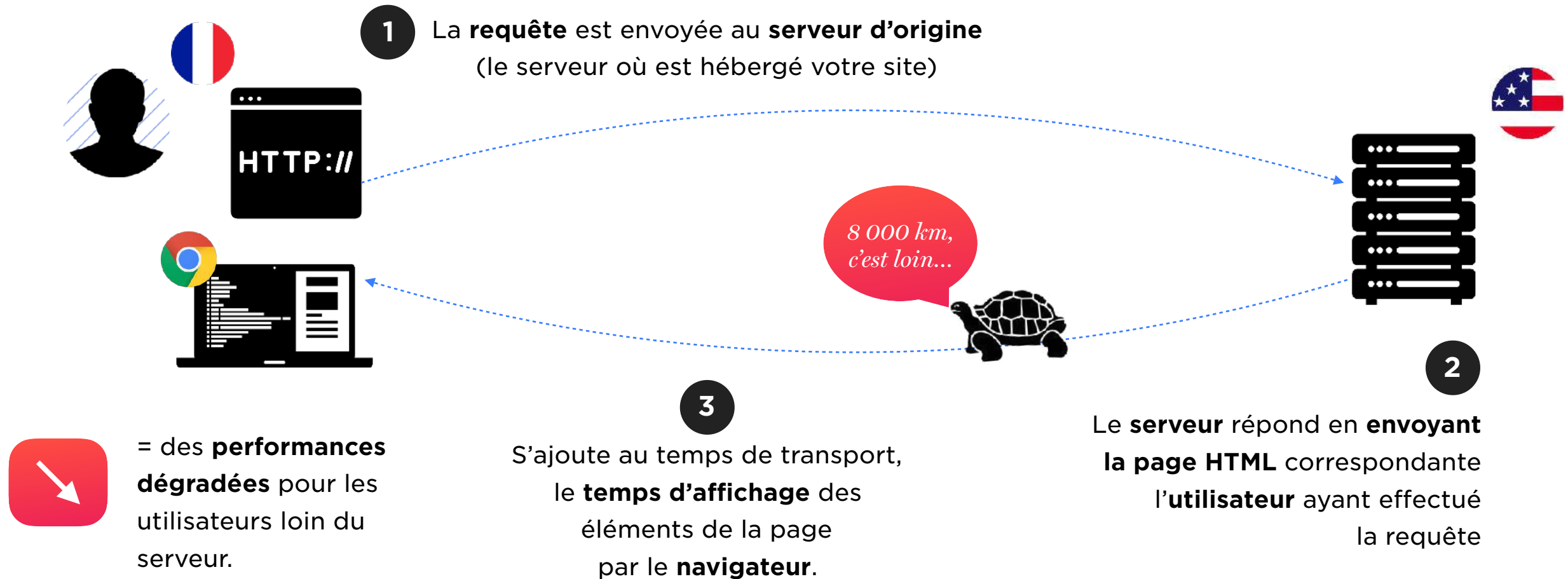
La notion de « Cache » sur un navigateur

Essentielle pour obtenir des performances optimales sur son site web & mobile



Fonctionnement d'une requête web

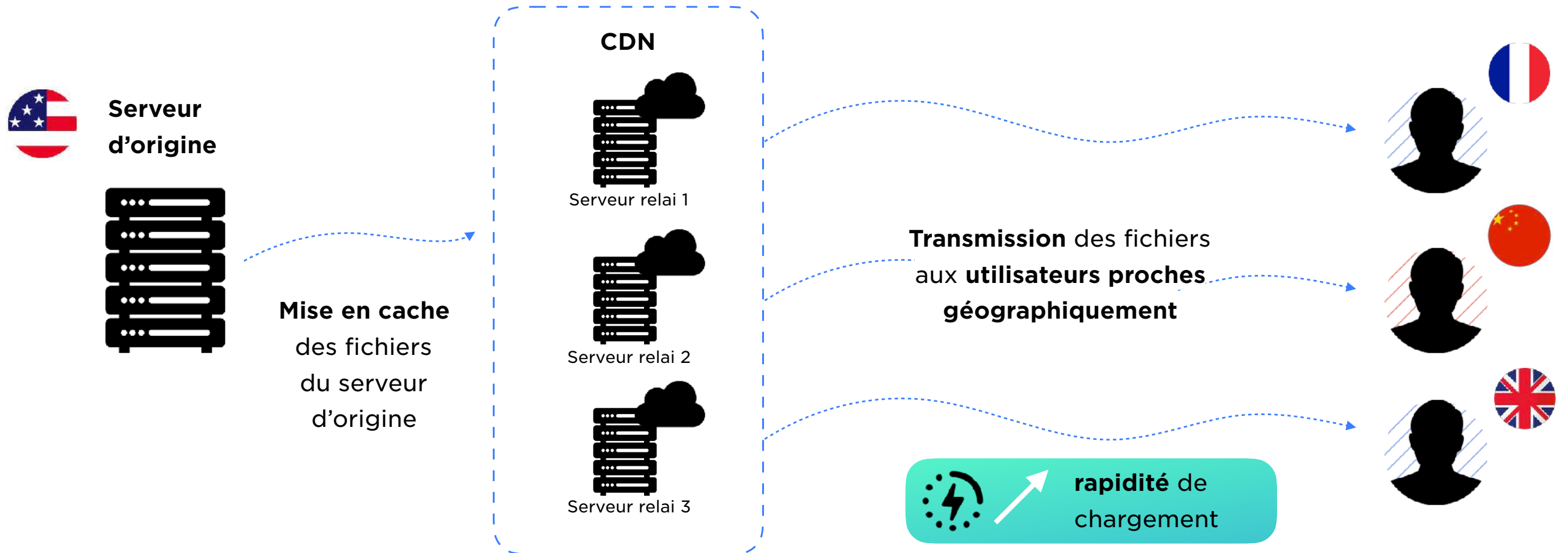
Lorsqu'on accède à une URL depuis un navigateur, que se passe-t-il ?



La notion de «CDN»

Pour éviter cette dégradation, on utilise des CDN.

Un CDN est un **réseau de serveurs répartis sur une zone géographique** : un pays, un continent, voire le monde entier. Ces réseaux vont agir comme **relais** entre le serveur d'origine et les utilisateurs finaux.



02

Analyser

Auditez votre site

Le temps de chargement ne sert à rien !

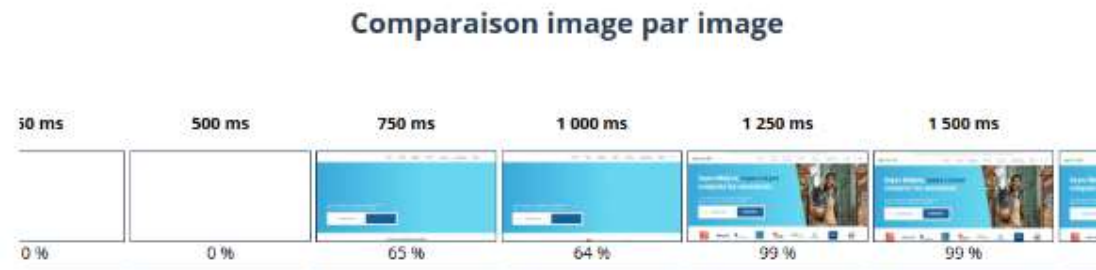
Les pages web se complexifient chaque jour et ne se résument plus à quelques lignes de code HTML. Toujours plus de médias, de fonctionnalités, de ressources externes... et, en parallèle, davantage de techniques de chargement progressif.

Dans un tel contexte, **le temps de chargement complet a donc perdu sa pertinence** pour évaluer la vitesse d'une page web **du point de vue de l'utilisateur**, et l'on doit ne lui trouver qu'un intérêt sur le plan technique.

Exemple

Dareboost a effectué un test sur les 2 sites concurrents lelynx.fr & lesfurets.com, déjà très performants. **Lelynx est le 1er à s'afficher. Pourtant**, lorsqu'on observe le temps de chargement complet de l'outil d'audit, **Lelynx affiche 45 secondes contre 5 secondes pour Les Furets !**

Le Lynx



User

1er
à s'afficher !



Chargement
complet

45
secondes

Les Furets



5
secondes

Preuve que le « temps de chargement complet » ne veut pas forcément dire meilleure performance : Lelynx affiche plus rapidement les éléments prioritaires à l'expérience utilisateur et fait du chargement asynchrone.

Nous allons voir dans nos bonnes pratiques, comment la performance implique des optimisations à différents niveaux. **D'où l'intérêt de mesurer plusieurs paramètres de votre site !**

Auditez votre site sous des indicateurs précis et mesurables

Il est nécessaire de s'appuyer sur différents **indicateurs** selon vos objectifs, les **techniques** que vous utilisez (chargement progressif, comportements asynchrones qui peuvent faire perdre de leur intérêt aux mesures de temps de chargement) et **utiliser des outils qui exposent clairement leur méthodologie** pour éviter les biais (type de connexion, d'appareil, de navigateur, d'emplacement par rapport au serveur...).



Dans l'ordre
de chargement :

Description

Comment le mesurer ?

First Byte

Temps de réponse côté serveur

Log serveur

First Paint (FP)

Moment où le premier pixel non blanc apparaît à l'écran

Webpagetest, Insight

First Contentful Paint (FCP)

Moment où le premier pixel non blanc apparaît à l'écran qui appartient à une image ou un texte

Webpagetest, Insight, Google Lighthouse

Fist Meaningful Paint (FMP)

Moment où la page est utilisable, où il est possible d'interagir

Webpagetest, Insight, Google Lighthouse

SpeedIndex

Index de référence provenant d'un algorithme open source qui calcule le temps moyen pour que les parties visibles au-dessus de la ligne de flottaison soient affichées.

Webpagetest, Insight, Google Lighthouse

Fully Load Time

Le temps complet pour charger tous les éléments de la page

Webpagetest

Time to Interactive (TTI)

Mesure des tâches longues, de la consommation CPU, du framerate, etc...

Javascript, Google Lighthouse

Custom Metrics

Permet de coder des événements à mesurer sur les particularités de votre site.

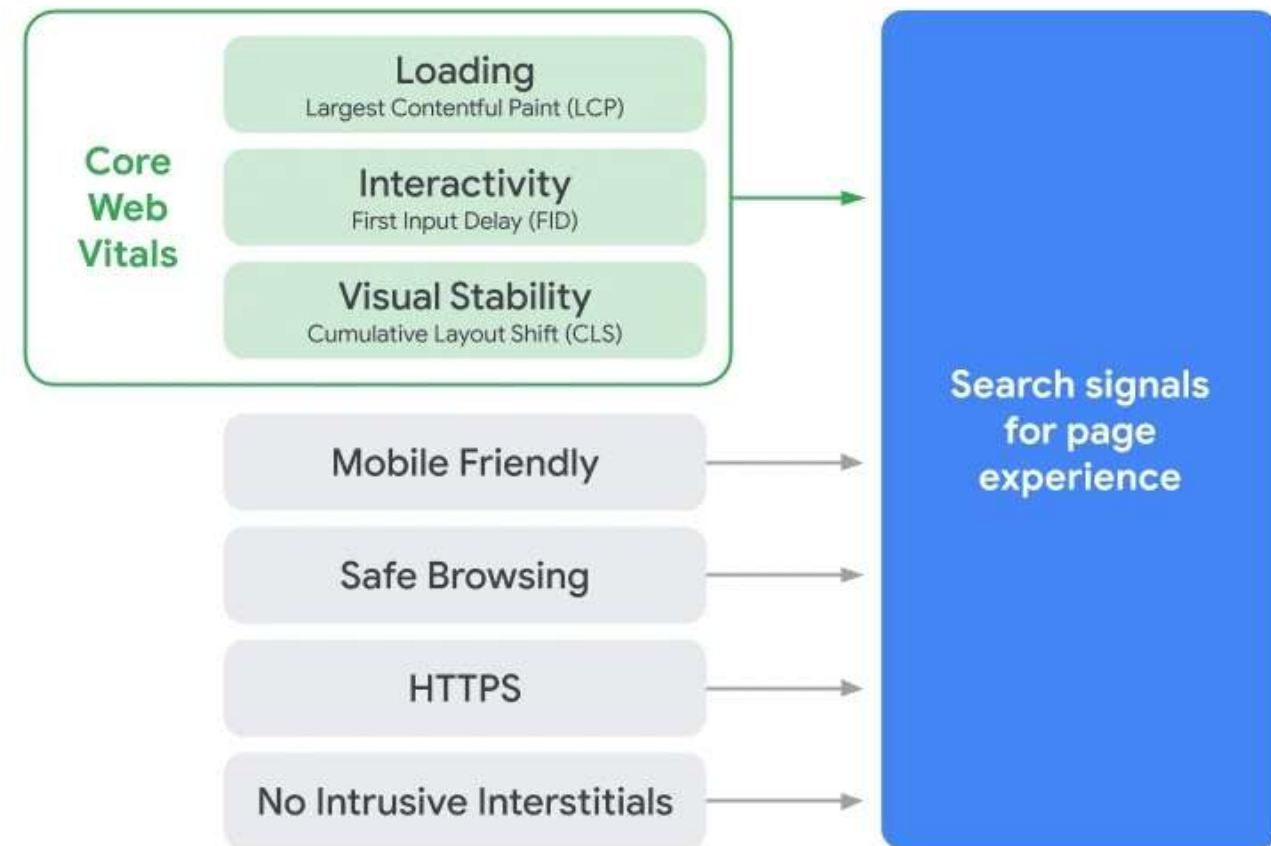
Javascript

Core Web Vitals : 3 nouveaux indicateurs à intégrer

Dès mai 2021, Google prendra en compte l'expérience vécue par l'utilisateur sur une page web dans ses résultats de recherche pour améliorer la vitesse et la qualité de l'expérience des sites.

Sept critères de mesure devront être respectés pour avoir une chance d'arriver en pole position des SERP. Quatre d'entre eux nous étaient déjà familiers, depuis 2015-2016 :

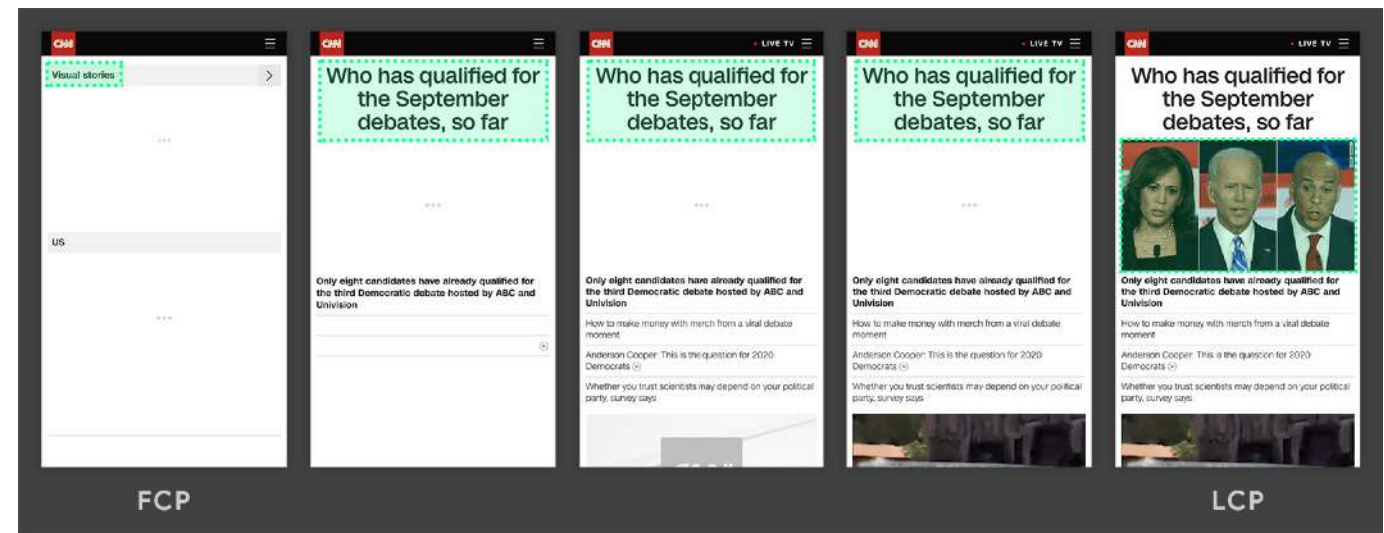
- **Mobile friendly** : cette page est adaptée à la navigation sur mobile
- **Safe browsing** : cette page ne contient pas de logiciels ou téléchargements malveillants pouvant nuire à votre sécurité...
- **HTTPS** : cette page est sécurisée via HTTPS
- **No intrusive interstitials** : cette page est accessible facilement, sans interstitiels qui appauvrissent l'expérience de navigation.



Core Web Vitals : 3 nouveaux indicateurs à intégrer

Trois nouveaux critères font leur apparition, les Core Web Vitals. Ils concernent :

- **Le chargement avec le Largest contentful paint (LCP).** Il calcule le temps nécessaire pour charger le plus grand contenu, en taille dans l'écran du viewport (donc sans scrolling de l'utilisateur). Que ce soit un paragraphe de texte ou une image, c'est vraiment l'occupation de l'espace de l'écran qui est prise en compte ici.



{Illustration de la mesure du LCP, ici la photo - Source : Google}

LCP

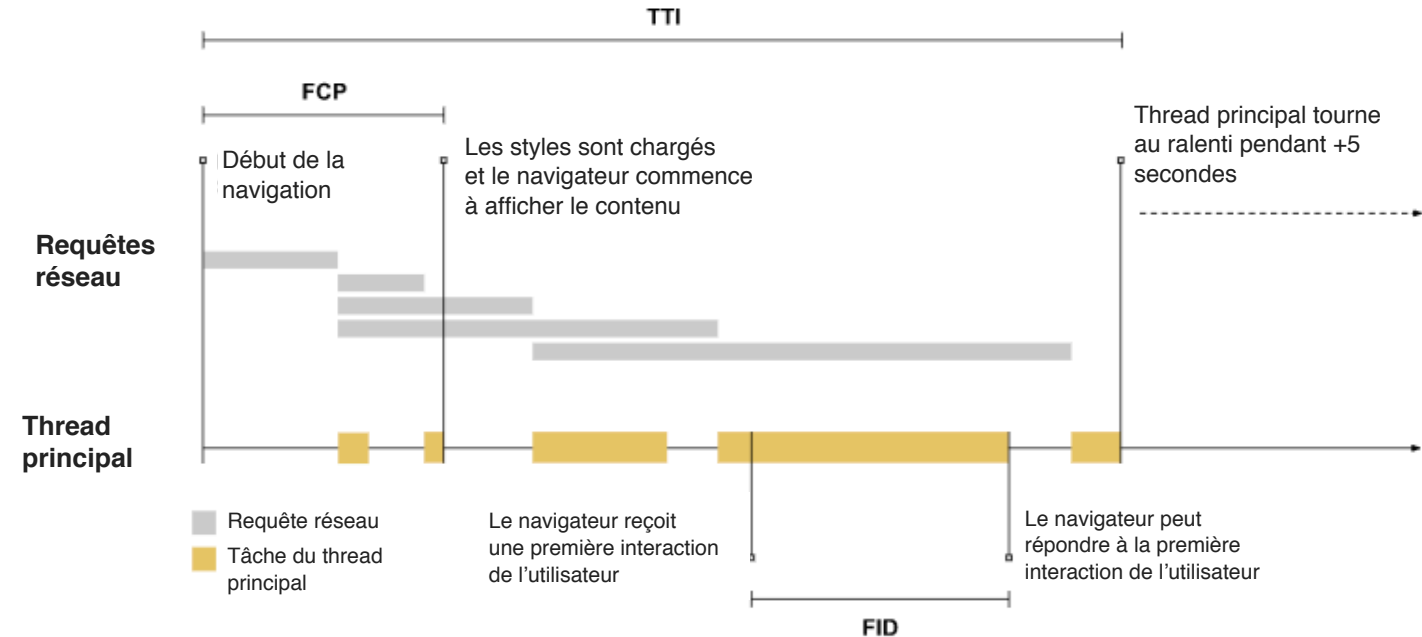
Largest Contentful Paint



Core Web Vitals : 3 nouveaux indicateurs à intégrer

- **L'interactivité via le First Input Delay (FID).**

Il mesure le temps entre la première interaction de l'utilisateur avec une page (clic sur un lien ou un bouton par exemple) et la réaction du navigateur à cette sollicitation.



FID

First Input Delay



{Illustration de la mesure du FID - Source : Google}

Core Web Vitals : 3 nouveaux indicateurs à intégrer

- **La stabilité visuelle via le Cumulative Layout Shift (CLS).** L'objectif de cet indicateur est de mesurer la part de contenu qui change de position dans le viewport et qui donne ce sentiment d'instabilité, de contenu qui « saute » et qui génère parfois de mauvaises interactions. Une expérience que l'on voit notamment sur les pages de certains médias avec de nombreuses bannières qui repoussent sans cesse le texte.



The impact fraction is 0.5 (taille : 50% of viewport)
the distance fraction is 0.14 (distance : 14% of viewport)
The layout shift score is $0.5 \times 0.14 = 0.07$



The impact fraction is 0.38 (taille : 38% of viewport)
the distance fraction is 0.3 (distance : 30% of viewport)
The layout shift score is $0.38 \times 0.30 = 0.1172$

{Illustration de la mesure du CLS - Source : Google}



CLS
Cumulative Layout Shift

La formule de Google consiste à multiplier la taille cumulée (en pourcentage du viewport) des éléments qui se déplacent dans la page, par la longueur maximum (toujours en pourcentage du viewport) de ce déplacement.

Google Lighthouse

Un outil **open source & gratuit** proposé par Google et intégré au navigateur **Chrome**.

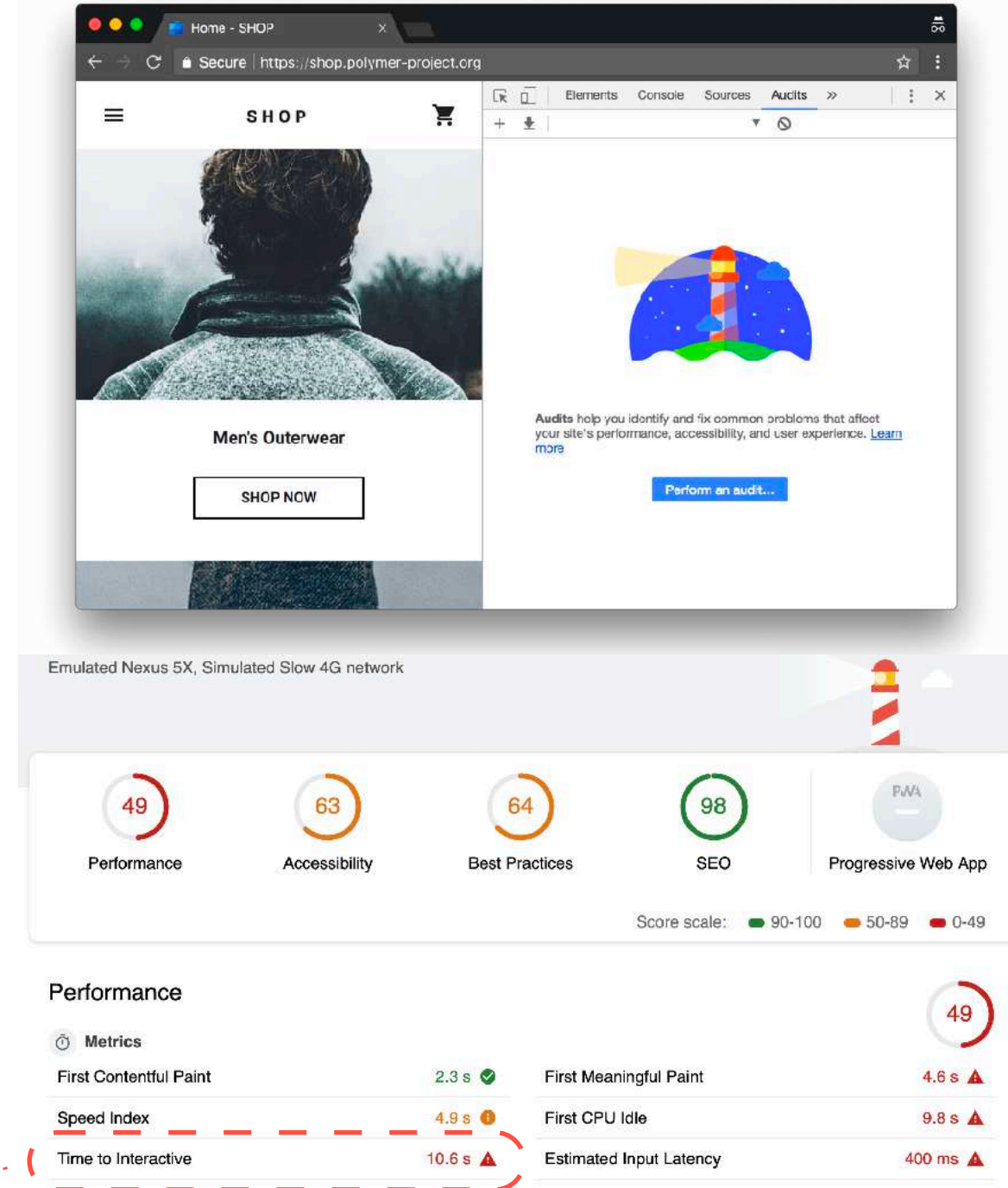
Plusieurs options :

- + Faire l'audit sur mobile ou desktop
- + Auditer les différents aspects de votre site (SEO, performances, accessibilité, respect des «bonnes pratiques »)
- + Simuler un type de connexion (slow 3G, Fast 3G)

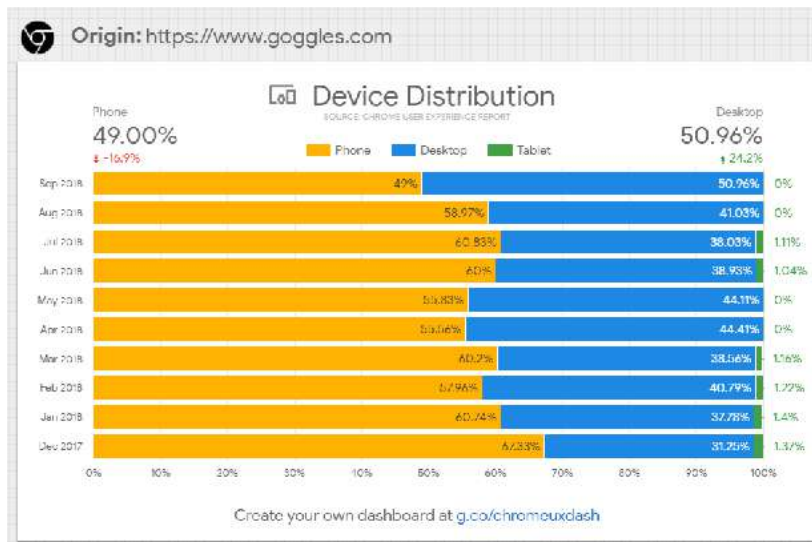
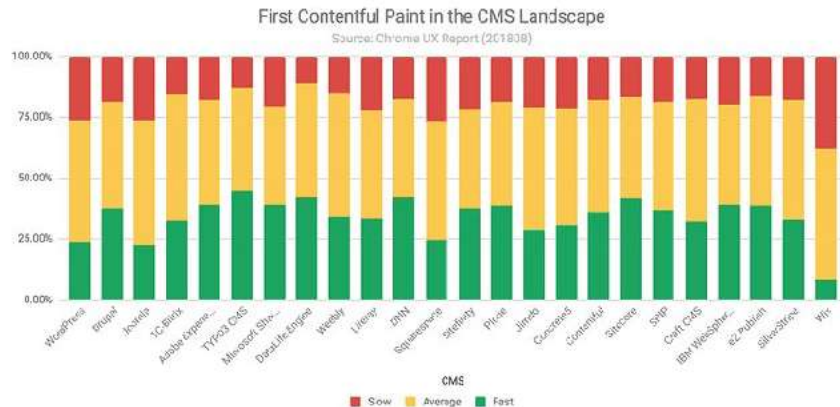
À l'issue de l'audit, l'outil donne un **score** à chaque catégorie auditée et affiche les principales **métriques** correspondantes.

Il affiche également le détail des points positifs et négatifs, ainsi que des **recommandations personnalisées**.

Par exemple, si le **TTI** est trop long, Lighthouse peut suggérer de compresser vos fichiers CSS & JS ou de ne charger que les images qui sont visibles par l'utilisateur.



Real User Monitoring



Grâce aux outils de BigData mis à disposition par Google, **il est possible de récupérer ces différentes valeurs vues de vos utilisateurs.**

L'outil **Chrome User Experience Report** vous permet d'accéder en BigQuery à toutes les données dans le temps de votre site... et de celles de vos concurrents !

Il existe aussi d'autres alternatives qui compilent l'ensemble des datas de vos utilisateurs :

- Pingdom
- AppDynamics
- New Relic
- Akamai

Palmarès - Evolution	Site	Speed Index	Temps de chargement des pages en millisecondes	Nombre moyen de requêtes	Poids moyen des pages analysées (octets)
1 (0)	Wikipedia	2 647	3 646	15	239 630
2 (0)	Google	3 349	5 500	25	585 158
3 (1)	Cdiscount	3 545	9 254	49	818 672
4 (-1)	franceinfo	3 550	8 698	93	613 957
5 (0)	Amazon	3 825	10 709	121	1 887 598
6 (1)	Free	3 947	5 934	36	587 816
7 (3)	Groupon	4 055	4 738	38	645 340
8 (-2)	Expedia	4 111	11 237	65	788 309
9 (-1)	YouTube	4 158	7 003	26	1 133 845
10 (-1)	Ouest France	4 210	6 438	25	437 553
11 (5)	La Chaine Météo	4 456	6 107	39	1 055 627
12 (1)	RTL	4 558	8 859	35	1 469 246
13 (4)	Fnac	4 792	20 091	172	1 339 761
14 (0)	Le Monde	4 887	16 751	154	1 626 014
15 (10)	AlloCine	4 997	4 696	122	1 121 662
16 (-5)	Marmiton	5 025	11 503	120	1 315 217
17 (-5)	Twitter	5 121	4 621	19	565 241
18 (-3)	Booking.com	5 343	10 770	50	884 697
19 (0)	Duck Duck Go	5 409	8 136	37	850 485
20 (-2)	L'Obs	5 638	11 738	78	1 367 685

Observez vos concurrents les + rapides

Une cinquantaine de sites parmi les plus visités sur mobile en France sont mesurés et **un classement est publié chaque mois sur le Journal du Net**. Rapidité d'affichage, légèreté des pages, bonnes pratiques en matière de webperformance, sont analysées via des indicateurs précis par Fasterize, partenaire du JDN. Trois pages distinctes sont scrollées sur chaque site mobile, plusieurs fois par jour, pour en ressortir des moyennes.

C'est le Speed Index qui classe les sites. Le ranking est également proposé par grands secteurs clés.

[Voir le classement](#)

03

Embarquer

Fixez des objectifs précis & collaboratifs

Adoptez une démarche de conception collaborative

Il est essentiel de sensibiliser les UX/UI designers aux problématiques de vitesse, d'où la nécessité de créer un « **budget performance** ».

Chaque partie prenante a des limites à respecter pour ne pas pénaliser la performance du futur site.

Des outils pour calculer un budget performance sont disponibles comme performancebudget.io.

Il suffit de saisir le temps de chargement que vous estimez nécessaire pour obtenir la meilleure expérience utilisateur puis le type de connexion sur laquelle surfe vos utilisateurs et l'outil calcule un budget performance.

Fixez votre « budget performance »

Qu'est-ce que c'est un budget performance ?

C'est un ensemble de contraintes que vous allez pouvoir appliquer à vos pages en fonction de votre cible et du matériel dont elle dispose.

Il s'agit d'une metric à respecter, par exemple :

- Poids page < 1 Mo
- Requêtes < 60 ressources (images, styles, scripts)

04

Optimiser

Nos conseils d'optimisation

En termes de dev, technique, UI & UX

Optimisez le poids de vos pages

Optimisez le code d'une page de manière à afficher en priorité ce qui se trouvera tout de suite visible à l'écran.

Home page



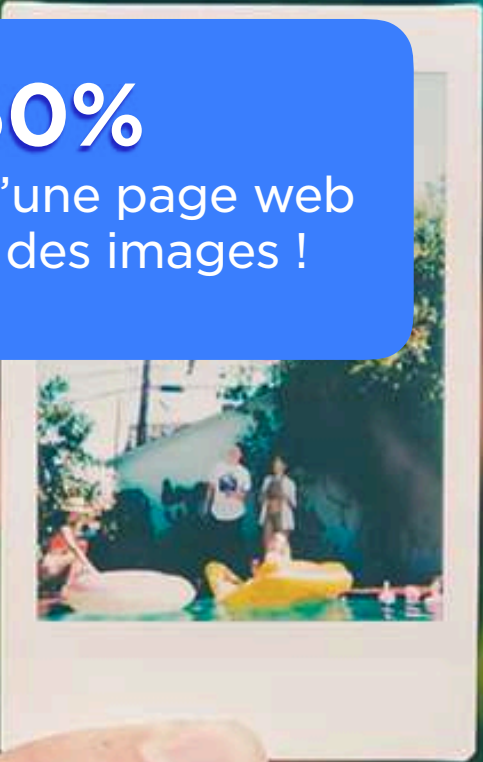
Connexion
3G



= 20 secondes pour se charger !

Guidelines

- + **Poids des pages :** 1 MB (optimal) - 2 MB max
- + **Nombre d'objets sur la page :** 40-50 (comprenant les images, fichiers de code - html, css, js, json - excluant les « 3rd party » contents)
- + **Poids des images :**
 - < 40 KB
(petites images)
 - < 120 KB
(moyennes images)
 - < 1 MB
(grandes images)



60%

du poids d'une page web
provient des images !

Optimisez vos images

Priorisez les images clés, optimisez leur poids, leur taille selon le contexte et utilisez des outils pour automatiser l'optimisation.

- + Choisissez le **bon format & la bonne taille**.

Par exemple, vous pouvez utiliser un outil UI pour voir le résultat d'une image réduite sans réduire la qualité d'affichage tel que Squoosh.

- + Pensez à l'**adaptation intelligente** des images aux tailles d'écrans. Par exemple, vous pouvez **mettre en place un système au niveau du CDN** pour automatiser la conversion et la compression des images (pour un terminal, un navigateur ou la taille de l'image demandée) tels que Cloudinary ou Akamai Image Manager.

Allégez vos animations : CSS 3

CSS 3, une alternative intéressante au Javascript :

- + Le temps d'exécution est généralement + rapide que pour du JS.
- + Vos animations seront visibles même si l'utilisateur a désactivé le JS sur son navigateur.
- + Il nécessite moins de lignes de code que le JS, ce qui contribue à alléger le poids du code de vos pages.

En revanche, il a quelques inconvénients :

- + Il est peu ou pas compatible avec les navigateurs les + anciens.
- + Le Javascript offre plus de possibilités d'animation (mais ce gap a tendance à se réduire).

Allégez vos animations : Lottie

Lottie est une solution open-source développée par Airbnb en 2017.

Elle permet de **convertir en fichiers JSON des animations créées dans After Effects** (de plus en plus utilisées par les designers pour animer illustrations vectorielles, logos et présentations de produits).

Les fichiers JSON sont des fichiers textuels très légers et sont donc beaucoup plus rapides à charger que des vidéos ou des javascript complexes.



Optimisez le chemin critique du rendu des pages

Qu'est ce que c'est ?

Ce que l'on appelle le chemin critique du rendu concerne ce qu'il se passe au cours des étapes intermédiaires entre la réception des octets HTML, CSS et JavaScript et le traitement nécessaire pour les transformer en pixels affichés à l'écran.

Il est nécessaire de définir pour chaque page l'ordre de chargement des éléments.

- + **Chargez les fichiers CSS avant les fichiers JS**, car les fichiers CSS sont utilisés pour l'affichage de la page, tandis que les fichiers JS servent pour l'interactivité de la page.
- + **Ne chargez que les fichiers CSS utilisés sur la page** (ne pas charger de fichiers CSS non utilisés).
- + **Minifiez** (compressez) les fichiers CSS & JS.
- + Utilisez les **attributs ASYNC et DEFER** pour vos appels Javascript : ils permettent de charger et lancer l'interprétation de code JavaScript sans bloquer le rendu HTML (son affichage à l'écran).

74%

du temps passé sur une page l'est sur les deux premières hauteurs d'écran*

7sec VS 3sec

on peut passer de 7 secondes à 3 secondes de temps de chargement d'une page grâce au lazy loading**



Privilégiez le lazy loading

Chargez les contenus **sous la ligne de flottaison à la demande**, lorsque l'utilisateur fait défiler la page.

Le premier chargement de la page n'est ainsi pas ralenti par des ressources non visibles par l'utilisation.



L'hébergement

Parfois sous-estimé, l'hébergement est pourtant un **élément important de la performance** de votre site.

Bien que l'architecture et le type d'hébergement puissent varier grandement en fonction des besoins de votre site, il existe des « bonnes pratiques » que vous pouvez appliquer.

Bonnes pratiques côté serveur

- + Optez pour un **serveur dédié** (si le trafic de votre site le justifie, par exemple si vous avez une centaine de visiteurs uniques journaliers).
- + Disposez d'un **serveur applicatif** (hébergement de votre site) et d'un serveur pour votre BDD séparés : les requêtes en base de données peuvent nécessiter beaucoup de ressources, en séparant votre BDD et votre hébergement, vous évitez que votre site ne soit ralenti par des grosses requêtes en base de données.
- + Mettez en place un **système de répartition de charge** (« Load Balancing ») afin de réduire le risque que votre site tombe lors des périodes de forte affluence.
- + Mettez en place un **système de cache type « Memcached »** qui permet de stocker les données dans la mémoire vive du serveur (accès plus rapide que de la mémoire physique).
- + Mettez en place un **réseau de CDN**
👉 voir slide 20.

Nous percevons un temps de chargement de **80 millisecondes supérieur** à la réalité
- Paul Bakaus

On observe une différence de plus de **30%** dans la perception de la vitesse entre des personnes pressées et des personnes calmes*



Pensez à la perception de la vitesse

Tous les utilisateurs n'ont pas la même perception de la vitesse, elle varie en fonction de :

- + L'objectif qu'il cherche à atteindre et sa perception d'importance,
- + Son âge,
- + Son état émotionnel (anxiété, calme, pressé...),
- + Son contexte : lieu, position, device d'utilisation...
- + L'**Effective Use**, c'est-à-dire le moment où un site semble effectivement utilisable.

Plusieurs leviers UX existent

Au-delà d'améliorer la performance technique pour un temps de chargement plus rapide, pensez à améliorer l'UX de votre site pour pallier la **perception** d'attente des utilisateurs.

Il existe **plusieurs leviers en conception** qui peuvent parfois paraître de petite envergure mais qui ont **toute leur importance** dans l'expérience.

Nos bonnes pratiques



Animez la phase de chargement

Affichez un petit jeu ou une petite animation interactive pendant un temps de traitement long est un bon moyen de garder l'utilisateur actif.

Utilisez la barre de téléchargement ou les spinners pour les temps de chargement les + courts.



Animez vos feedbacks d'actions

Animez vos feedbacks pour montrer que le système traite la demande de l'utilisateur.



Utilisez les fonctionnalités qui gagnent du temps

Ne négligez aucun moyen de gagner du temps : autocomplétion, historique, Touch ID / Fingerprint, remplissage automatique des champs par le navigateur, choix du bon clavier en mobile...

04 Optimiser

Zoom sur les solutions du marché

Site dédié, responsive, adaptive, AMP, PWA, PWAMP...

Sites mobiles

*Les sites dédiés au mobile avec une url
"m.domaine.com"*

Les sites mobiles



Un site mobile dédié est **distinct** de votre site web pour ordinateur.

Le système détecte que les internautes se connectent depuis un mobile et les redirige vers le site optimisé pour mobile (généralement par le biais d'un sous-domaine tel que [m.votredomaine.com](#)).

Ainsi, seuls les mobinautes verront le **site dédié au mobile**.

Les sites mobiles



Avantages

- + Ils permettent d'avoir une expérience mobile différente de la version desktop (navigation & contenu différents).
- + Ils permettent d'utiliser les fonctionnalités du support mobile (appel, appareil photo, etc.).
- + Ils ont de meilleures performances qu'un site responsive, car seul le code et les médias dédiés au mobile sont chargés.
- + Comme ils ne sont pensés que pour le mobile, leur navigation est optimisée pour le tactile.



Inconvénients

- + Plusieurs URL ce qui nécessite de faire attention aux redirections entre la version mobile et les autres versions pour une totale cohérence.
- + Complexification du référencement sur les moteurs de recherche du à ces URL distinctes. Les annotations sont importantes pour que les Googlebots détectent le contenu et que les algorithmes comprennent la relation entre vos pages pour ordinateur et celles pour mobile.
- + Duplication du contenu & de l'infrastructure : le fait d'avoir deux ensembles de contenus distincts peut rendre la gestion des données plus complexe et coûteuse.

Responsive Design

*Une seule url, un code, un affichage
qui s'adapte*

Le responsive design, c'est quoi ?



Le « Responsive Design » est une pratique qui consiste à créer un site unique dont le **design** va **s'adapter à l'appareil** de l'utilisateur grâce à système de grilles fluides, afin de lui offrir une ergonomie optimale en respectant la qualité de lecture et de navigation du site, quel que soit l'appareil sur lequel il se connecte (mobile, tablettes, desktop, etc.).

La base de **code HTML** est **commune** à tous les terminaux, seule la mise en page change, pas le contenu, il est néanmoins possible de cacher un contenu visible sur le site desktop pour empêcher son affichage sur mobile.

Le responsive design



Avantages

- + Une URL unique qui s'adapte à tous terminaux : contenu identique (HTML), mais design différent (CSS).
- + Une expérience utilisateur simplifiée grâce au paramétrage du contenu et des fonctionnalités qui peut être opéré selon le device.
- + Moins coûteux, car une seule version du site a développer et a mettre à jour.
- + Un seul back office pour gérer votre contenu.
- + Un meilleur référencement opéré par les moteurs de recherche, un SEO facilité.



Inconvénients

- + Votre site mobile est lié à votre site desktop : si l'un tombe, l'autre aussi.
- + Des performances pas toujours au rendez vous : le temps de chargement n'est pas optimal du fait que l'intégralité du code soit chargé (mobile, tablet & desktop) même s'il n'est pas affiché.
- + Moins de personnalisation que dans un site adaptive, seule la mise en page change réellement.

Dynamic Serving

*Une seule url, plusieurs codes,
un site personnalisé par terminal*

Le dynamic serving, c'est quoi ?



Le Dynamic Serving (aussi appelé diffusion dynamique ou adaptive design) est une configuration dans laquelle le serveur diffuse un **code HTML & CSS différents** sur la **même URL** en fonction du **device** qui requiert l'accès à la page.

La différence majeure avec le Responsive concerne l'utilisation de **codes distincts** pour chaque appareil (smartphone, desktop, tablette, TV connectée...) assurant une personnalisation plus poussée.

Le dynamic serving



Avantages

- Une expérience unique et personnalisée pour chaque device.
- Une URL unique pour tous les devices qui facilite le référencement par les moteurs de recherche.
- Performance optimisée car on ne charge que le code et les médias liés à l'appareil détecté (desktop ou mobile ou tablette).



Inconvénients

- Plus coûteux car plusieurs structures de site à développer et maintenir, complexité des mises à jour de contenu commun à toutes les versions.

AMP

*Une version HTML AMP propre à Google
et un ensemble de guidelines pour un site
très light et rapide*

AMP, c'est quoi ?

Accelerated **M**obile **P**ages est un projet open source lancé en 2015 par Google qui permet d'afficher une page web sur mobile beaucoup plus rapidement.

Comment ça se matérialise ?

Dans la page de résultats de recherche, les sites AMP apparaissent dans un carrousel (pour les sites d'actualités) avec une pastille & mention AMP. À partir de mai 2021, cette espace de mise en avant sera ouvert à des contenus non AMP, à condition de respecter la politique de Google Actualités.

Lorsque l'utilisateur sélectionne un site depuis les résultats de recherche, la page se charge quasi instantanément (moins de 2 secondes).



AMP : Une expérience + engageante

En moyenne,
1 sec

de temps de chargement

Des pages
4 fois

+ rapides

10 fois moins
de données mobiles

15 à 85%
de gain de vitesse
d'affichage

par rapport aux pages non AMP

Google valorise les contenus AMP

5,6%

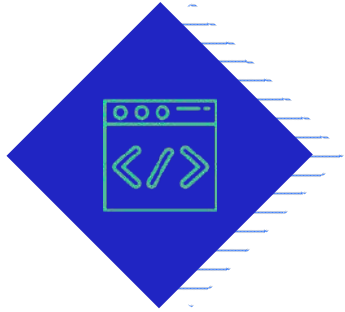
Une page
qui adopte AMP
améliorerait son
positionnement
Google de 5,6%
selon Condé Nast

En créant des articles AMP, les éditeurs / marques voient leurs contenus **mis en avant directement sur la page de résultats de recherche Google.**

Les pages AMP, bien visibles dans les SERP de Google, incitent davantage au clic. Selon le vice-président et spécialiste SEO de Condé Nast, **le CTR d'une page AMP augmente de 2%** comparé à une qui n'est pas adaptée à ce format.

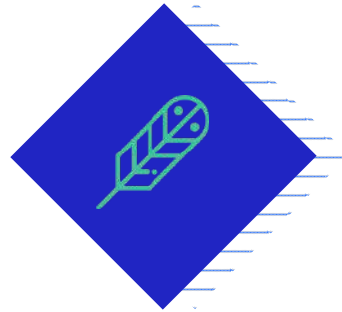
Ainsi, en favorisant le taux de clic, une page AMP sera privilégiée par Google en lui accordant un meilleur positionnement qu'une page moins cliquée.

Comment ça fonctionne ?



HTML AMP

Avec des pages AMP, lorsqu'un contenu est consulté depuis les résultats de Google, c'est sa version AMP qui est systématiquement appelée à la place de la page HTML classique.



TRÈS LIGHT

Il s'agit en fait de pages HTML très légères, auxquelles s'ajoutent le système de cache CDN et une réduction du Javascript.



PRIORISATION

L'image principale et le texte sont chargés en premier, au lieu de la publicité et des scripts en temps normal.



LECTURE AU SEIN DU DOMAINE GOOGLE

L'internaute qui sélectionne un contenu AMP n'est en fait pas redirigé vers le site à l'origine du contenu : la lecture se fait directement sur le domaine de Google, les pages AMP sont stockées en cache sur les propres serveurs de Google.

3 parties composent l'AMP

AMP HTML

Une déclinaison du HTML comportant certaines restrictions pour assurer des performances fiables, ainsi que certaines extensions permettant de créer directement du contenu riche sans javascript ou CSS complexe.

BIBLIOTHÈQUE AMP JS

Elle garantit un rendu rapide des pages AMP HTML et permet d'intégrer, dans des pages AMP, des composants dynamiques et asynchrones tels que les analytics, les frameworks de publicités, et les tunnels ecommerce.

AMP CACHE

Il fournit les pages AMP HTML sur un réseau CDN ultra rapide permettant de gagner en temps de réponse sur des data-center répartis sur le monde entier.

AMP : Avantages & Inconvénients



Avantages

- + Des gains très importants sur la vitesse de chargement des pages.
- + Une base technique simple : HTML5 + CSS3 + JS + cache Google.
- + L'affichage de l'URL du site, jusqu'à présent les pages AMP possédaient des bases d'URL plutôt longue avec la mention AMP et Google.
- + Un boost de visibilité grâce à l'affichage des pages AMP (articles et AMP Stories) dans un carrousel dédié au cœur des SERP (mais pas un critère de ranking pour le SEO).



Inconvénients

- + Un projet relativement lourd puisque c'est une nouvelle version d'un site à mettre en place.
Par exemple, une version responsive et un ensemble de pages AMP vont cohabiter.
- + Des restrictions techniques même si les mises à jour sont fréquentes : codage strict, pas de data third party, d'habillage publicitaire ou d'interstitiels, paywalls non pris en charge... la monétisation est un problème de l'AMP.

Que faut-il faire en AMP ?

Plutôt que de refaire son site mobile entièrement en AMP, **certaines pages peuvent l'être, d'autres non.**

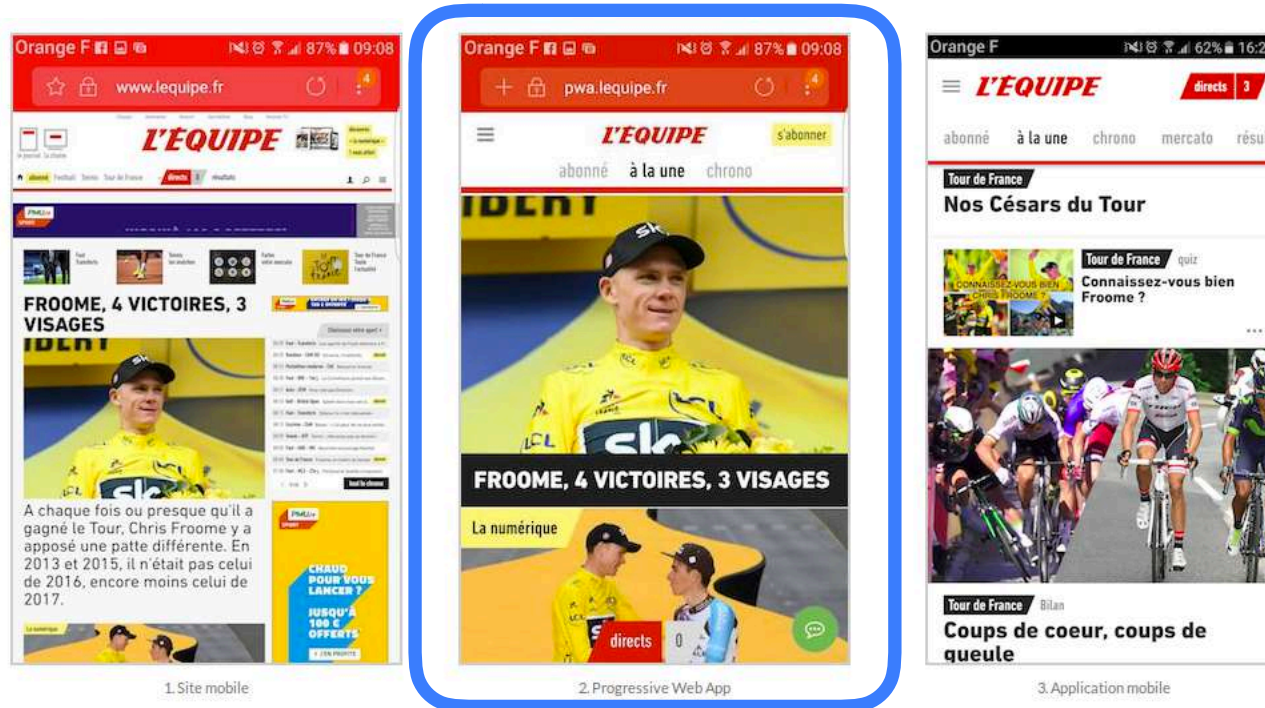
Tout dépend de votre activité et de vos pages de destination les plus importantes.

Par exemple, les fiches produit pour proposer un accès rapide au produit dès la page de résultats, ou encore une page liste.

PWA

*Une web app qui s'inspire des fonctionnalités
d'une app native*

Une PWA n'est pas une app native



Les PWA sont une évolution du web, elles ne sont donc **pas uniquement sur mobile**.

Il s'agit d'une **web-app** qui s'inspire des fonctionnalités natives d'une app mobile (logicielle). A ne pas confondre avec une app mobile native ou une app hybride !

Les PWA fonctionnent sur **iOS** (sans passer par l'AppStore) & **Android** (disponibles depuis peu sur le Google Play Store via TWA) et n'impliquent pas de refaire tout son site, mais d'ajouter des briques.

Les spécificités de PWA



+ être **ajoutée** à
l'**écran d'accueil**
comme une appli



+ fonctionner
hors connexion

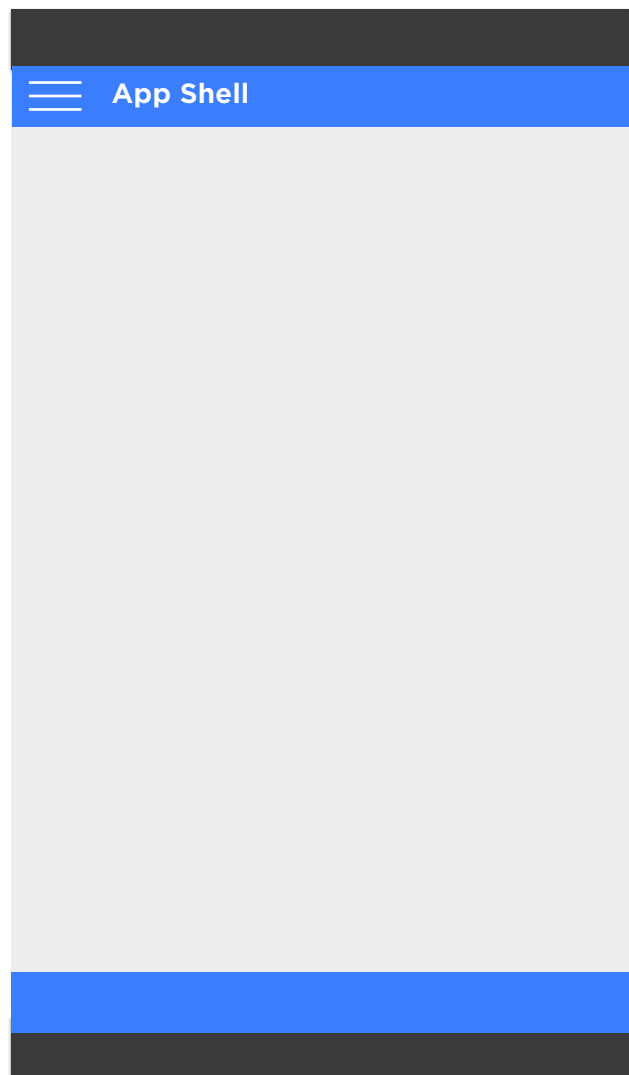


+ **envoyer des notifications**
et permettre de
meilleures performances,
plus rapides via une
gestion optimisée du
cache



+ permettre une
inscription simplifiée
(one tap sign-up & auto
sign-in)

3 éléments clés de PWA : l'App Shell



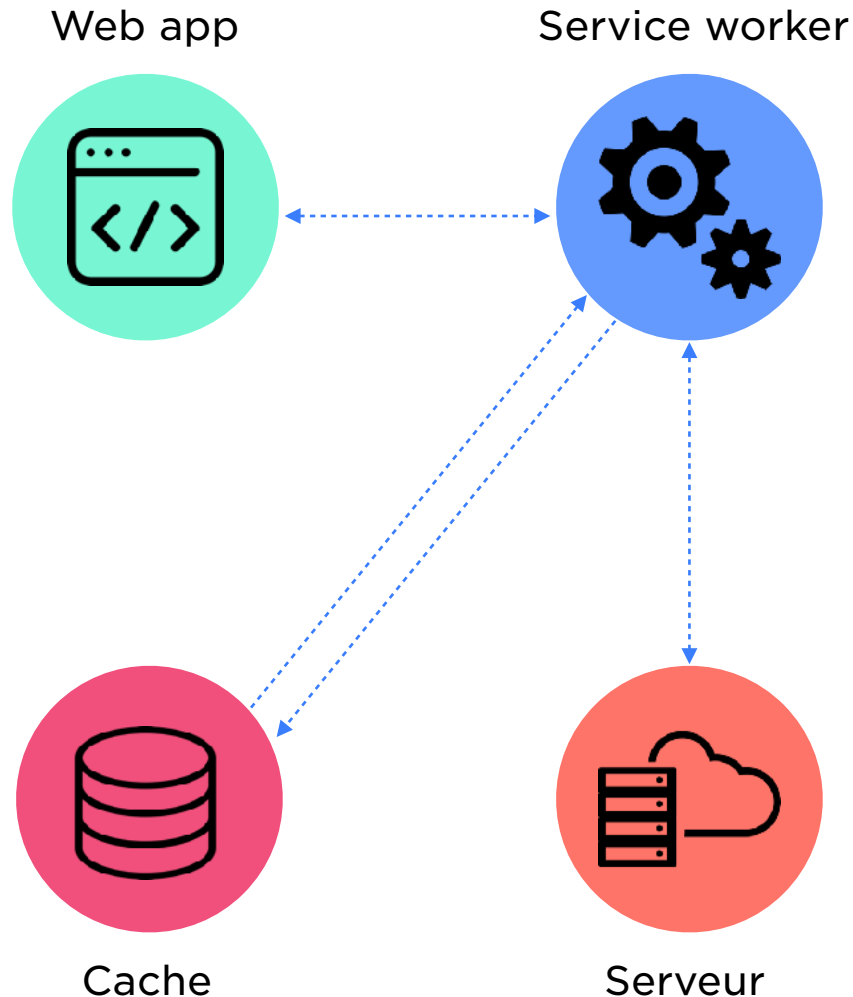
« L'app shell » (par opposition à « l'app content ») est le **squelette d'une PWA**.

Il contient les éléments principaux d'interface et les composants strictement nécessaires pour le fonctionnement de l'interface utilisateur (ex : le menu et les icônes liés à la navigation.)

Ces éléments sont conservés localement et seront servis (à partir du deuxième passage dans l'application) **depuis le cache de l'application**.

Ainsi, lors de la navigation seul le contenu de la PWA sera chargé dynamiquement (via une API), ce qui va permettre d'obtenir des **temps de chargement très courts**, tout en limitant la consommation de la bande passante.

3 éléments clés de PWA : service workers



Un service worker est un **fichier javascript exécuté par un navigateur**. Il sert **d'intermédiaire entre la web app et le navigateur** (ou le réseau, si celui-ci est disponible), et effectue des actions en réponse à des requêtes réseau ou des infos transmises par le système d'exploitation (ex : passage en mode « hors-connexion »).

Il permet notamment aux développeurs d'utiliser des fonctionnalités avancées telles que :

- + **l'envoi de notifications push**
- + **la gestion du cache**

C'est cette dernière particularité qui va permettre d'améliorer l'expérience utilisateur en stockant les données d'une page en cache.

Par ailleurs, il s'exécute en arrière plan, ce qui veut dire qu'il effectue les tâches qui lui sont assignées même si l'utilisateur ne navigue pas sur le site.

3 éléments clés de PWA : le Manifest

```
{
  "short_name": "Maps",
  "name": "Google Maps",
  "icons": [
    {
      "src": "/images/icons-192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/images/icons-512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": "/maps/?source=pwa",
  "background_color": "#3367D6",
  "display": "standalone",
  "scope": "/maps/",
  "theme_color": "#3367D6"
}
```

Le « **Manifest** » est un simple fichier **.json** qui va indiquer à l'appareil comment la PWA doit se comporter une fois installée sur votre appareil. C'est ce fichier qui va permettre à votre application d'avoir un rendu proche d'une application native et de la rendre accessible depuis votre écran d'accueil.

Il contient plusieurs informations (« métadonnées ») qui indiquent :

- + Le **nom** de votre PWA
- + Son **icône** (une fois ajoutée à l'écran d'accueil)
- + Le fait qu'elle doit se jouer en **plein écran** (sans la barre du navigateur) comme une application native.
- + Son **URL** : c'est ce qui lui permettra d'être accessible depuis l'écran d'accueil de votre mobile.

Une meilleure intégration sur Android

Très bien intégrée à Android, voire considérée comme une app classique, une PWA peut :

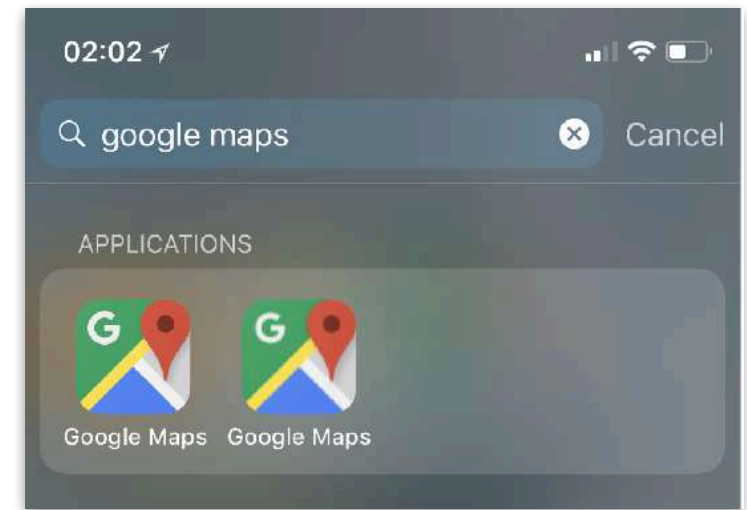
- + Être appelée à partir d'autres apps
- + Stocker plus de 50 Mb
- + Utiliser le Bluetooth
- + Avoir recours à la reconnaissance vocale
- + Envoyer des push notifications
- + Permettre le sign-in automatique et le one tap sign-up (connexion via un token, pas besoin de mot de passe)

Plus limité sur iOS

L'accès à la géolocalisation, la caméra, les capteurs (magnétomètre, accéléromètre...), la sortie audio, la synthèse vocale ainsi qu'Apple Pay est possible.

En revanche, contrairement à une app classique, **certaines fonctionnalités ne sont pas accessibles** : Bluetooth, numéro de série, Beacons, Touch ID, Face ID, ARKit, altimètre, batterie. La PWA **n'accède pas non plus aux informations personnelles** (contacts, localisation en background), appli de réseaux sociaux, paiements in-app, et bien d'autres services d'Apple.

Les push notifications ne sont pas non plus actionnables, ni l'intégration de Siri. Par exemple, si l'utilisateur demande à Siri d'ouvrir une PWA, il ne la trouvera pas.



PWA varie selon les OS & navigateurs

Chrome

	MOBILE		DESKTOP		
	Android	iOS	Windows	macOS	Linux
PWA Progressive Web Apps	✓	✓	✓	✓	✓
⚙ Service Worker	✓	✗	✓	✓	✓
🔔 Web push notifications	✓	✗	✓	✓	✓
📶 Offline browsing	✓	✗	✓	✓	✓
🔄 Background synchronization	✓	✗	✓	✓	✓
🏠 Add to home screen	✓	✗	Soon**	Soon**	--
🚀 App launching screen	✓	✗	--	--	--
📶 Bluetooth	✓	✗	✗	✓	✗
📶 Beacons*	✓	✗	✗	✓	✗
📍 Geolocation	✓	✓	✓	✓	✓
📍 Geofencing	✗	✗	✗	✗	✗
📷 Image capture	✓	✓	✓	✓	✓
📹 Video capture	✓	✓	✓	✓	✓

Safari

	MOBILE		DESKTOP		
	Android	iOS	Windows	macOS	Linux
PWA Progressive Web Apps	✓	✓	✓	✓	✓
⚙ Service Worker	--	✓	✗	✓	--
🔔 Web push notifications	--	✗	✗	✓	--
📶 Offline browsing	--	✓	✗	✗	--
🔄 Background synchronization	--	✗	✗	✗	--
🏠 Add to home screen	--	Soon**	--	--	--
🚀 App launching screen	--	Soon**	--	--	--
📶 Bluetooth	--	✗	✗	✗	--
📶 Beacons	--	✗	✗	✗	--
📍 Geolocation	--	✓	✗	✓	--
📍 Geofencing	--	✗	✗	✗	--
📷 Image capture	--	✓	✗	✓	--
📹 Video capture	--	✓	✗	✓	--

PWAMP

*Allier les deux solutions pour encore
plus de performance*

PWAMP : la combinaison des deux pour une meilleure performance

Ces deux formats sont faits pour travailler ensemble dans la mesure où ils se complètent parfaitement afin d'obtenir **une expérience mobile plus complète et plus adaptée.**

Par la rapidité de chargement de l'AMP et l'affichage 100% mobile friendly du tunnel de conversion du PWA, on aboutit à un site mobile rapide sur lequel on peut finaliser sa commande en quelques clics.



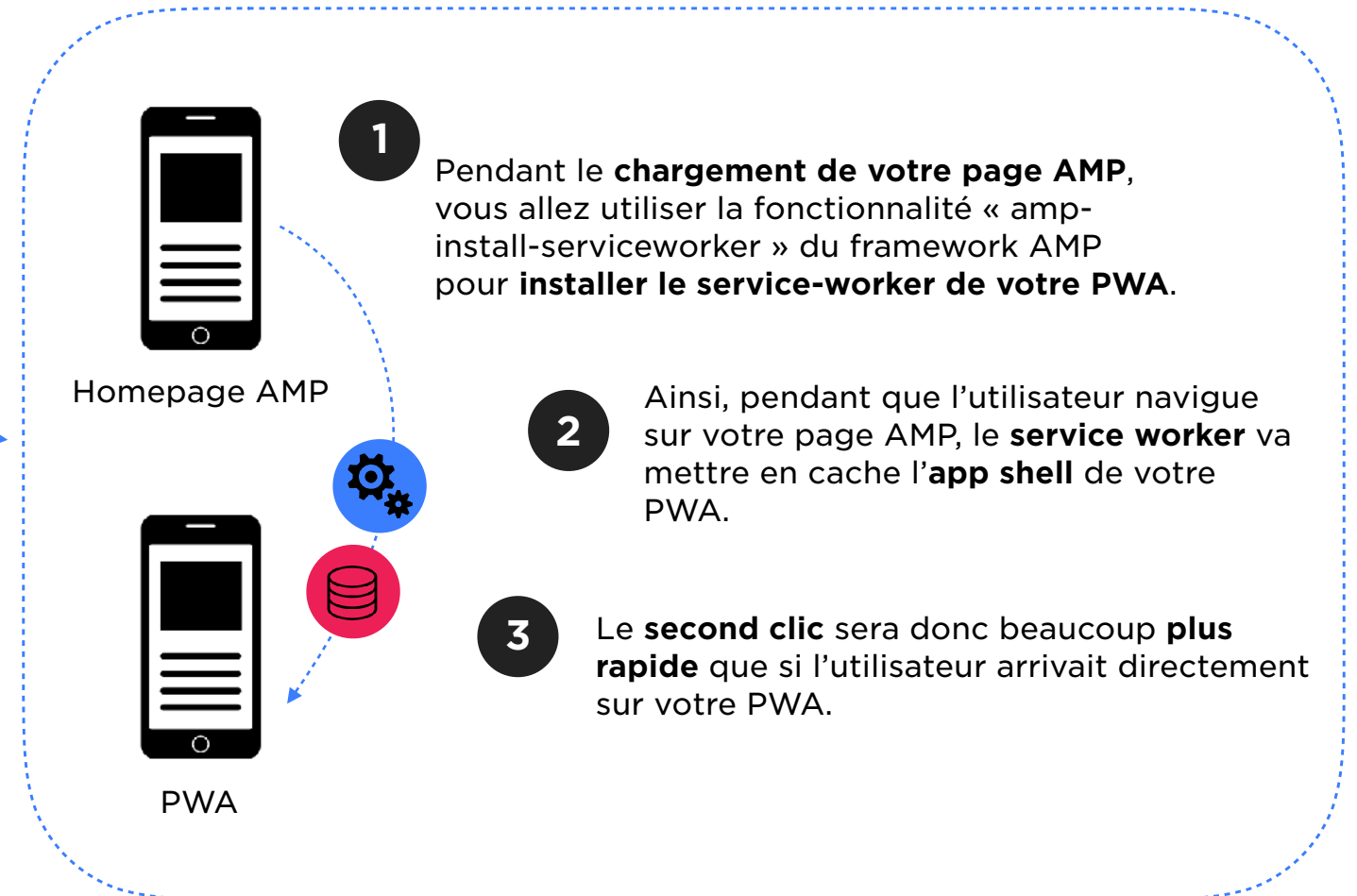
PWAMP : zoom sur AMP Up



Sur une PWA le **chargement initial est long** : le « service worker » (qui gère la mise en cache » de la PWA) n'est actif qu'à partir de la 2^{de} utilisation.

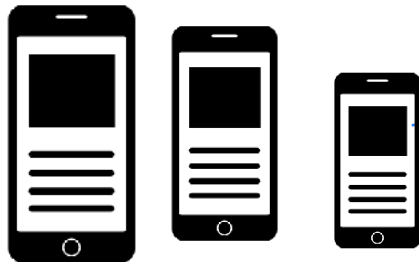


Vous pouvez utiliser **AMP** pour créer les **points d'entrée** de votre site (homepage, page produit...) qui vont pointer vers la PWA.



PWAMP : zoom sur AMP Down

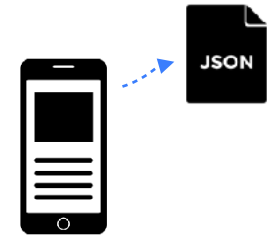
Si vous disposez déjà de **pages AMP** sur votre site, mais pas d'une PWA, vous pouvez réutiliser vos pages AMP pour **accélérer le développement de votre PWA**.



Pages AMP



- 1 En utilisant la fonctionnalité « **AMP Shadow** » vous pouvez **charger le framework AMP** dans **votre PWA**.
- 2 Cela va vous permettre d'utiliser vos **pages AMP** comme **source de données** pour votre PWA.
- 3 Lorsqu'un utilisateur clique sur un **lien au sein de votre PWA**, ce sera donc la **page AMP** qui lui sera affichée.



Dans la plupart des cas, une **PWA** se connecte à une **API JSON** en Ajax afin de récupérer les **données** qui vont venir **nourrir** votre **PWA**.

Besoin d'accompagnement pour améliorer la performance de votre site ? ⚡ 📱

Audit technique, conseil & stratégie, mise en place de *quick win*, refonte...
Contactez-nous uainfo@useradgents.com

USERADGENTS

Notre accompagnement « Design for speed »

Conception

Analyser les performances de votre site et définir les meilleurs choix techniques au prisme de votre écosystème.
Accompagner vos équipes dès la conception.

Expertises :

- + Audit technique
- + Conseil technologique
- + API management
- + UX
- + UI

Développement

Réaliser un développement sur-mesure par des équipes in-house.

Langages :

- + PHP
- + Node.js
- + Angular.js
- + Vue.js
- + React.js

Maintenance

Maintenir les solutions mises en place avec un centre de services.
Suivi des développement avec une TMA ajustée aux besoins clients.

Outils :

- + Atlassian : pour synchroniser les métiers & agréger le contenu
- + Nevercode : pour une intégration continue
- + SonarQube : pour un contrôle qualité continu

Marie BILLON

Chargées d'études digitales @ Trendwatchers

m.billon@useradgents.com

Solange DERREY

Directrice du Pôle Trendwatchers & de la
Communication

s.derrey@useradgents.com

Robin LODEON

Chef de projet technique
r.lodeon@useradgents.com



USERADGENTS

8 rue de la Rochefoucauld - 75009 Paris

01 77 75 65 90

www.useradgents.com