

Performance Web

surveiller et optimiser

<https://boris.schapira.dev/20190611-node>





**Boris
Schapira**



w e  s p e e d



Comprendre

Collecter & Automatiser

Savoir-faire, faire savoir

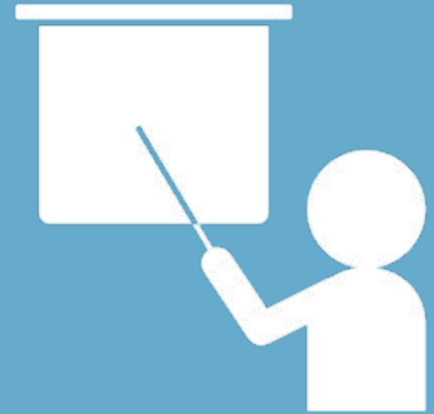
Maintenir



X ms → X %

X ms → X %

X ms → X %





Un retour d'XP
est une **histoire**
pas une **promesse.**



SITES LENTS

!?!

MAUVAISE UX



**« Speed isn't just a
feature, it's the
feature »**

« The Google Gospel of Speed »

Urs Hoelzle, 2012

Et depuis, la demande ↗



- Expérience *digitales*
- Fragmentation des contenus
(POSSE, réseaux sociaux)
- vs. Agrégation des contenus
(PESOS, réseaux de (re)information)
- Écoconception et écoréalisation

Alors la technique ↗



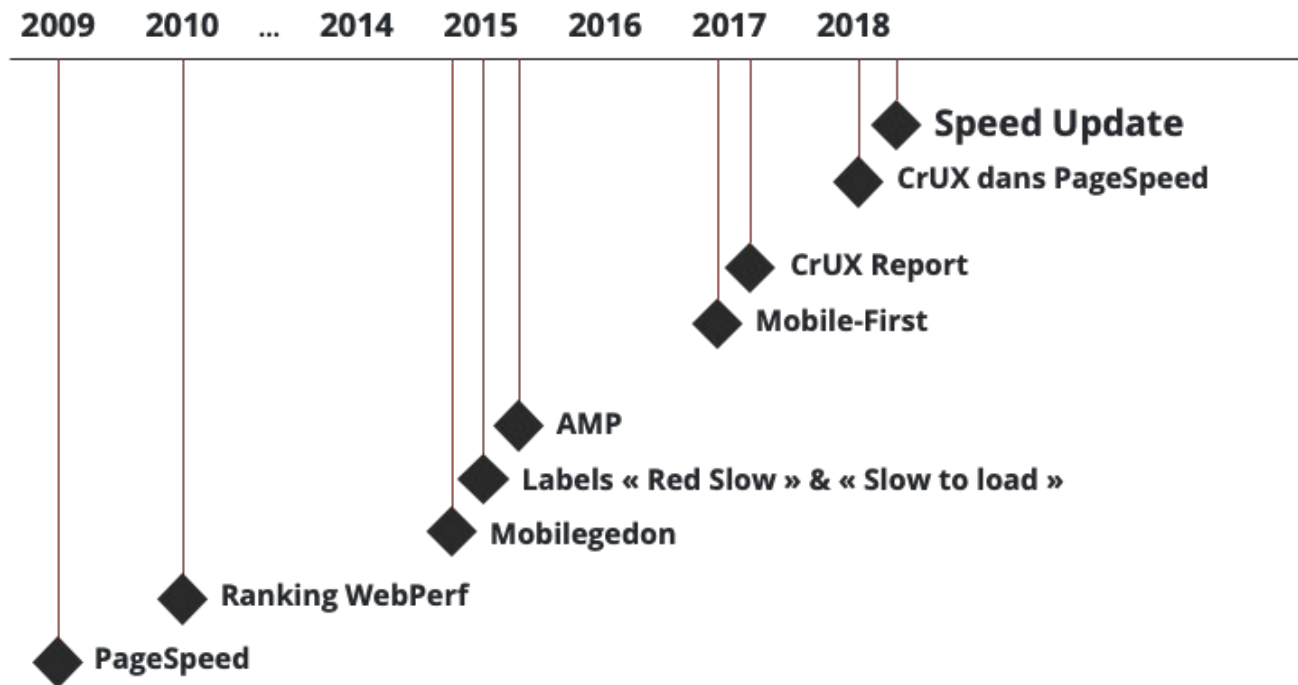
- Optimisation dans la diffusion des contenus (CDN)
- Nouvelles architectures (microservices, serverless)
- Maîtrise des mise-en-page adaptatives
- Gestion de formats d'image suivant les besoins
- SPA pour détruire le concept de “page”

Tout le monde aime la WebPerf



-  Coût d'exploitation (Réseau, CPU)
- Meilleure UX  meilleur SEO
- Augmentation des gains publicitaires
- Pour les propriétaires comme pour les entreprises du Web

Google, WebPerf, 2009-2018





Dans un monde idéal...

... nous voudrions savoir quand un·e utilisateur·ice :

1. a la confirmation que le chargement **a commencé**
2. a l'impression, **en regardant la page**, qu'il peut interagir
3. **peut réellement interagir** avec la page
4. peut interagir **sans délai et sans ralentissements**

De nombreux indicateurs

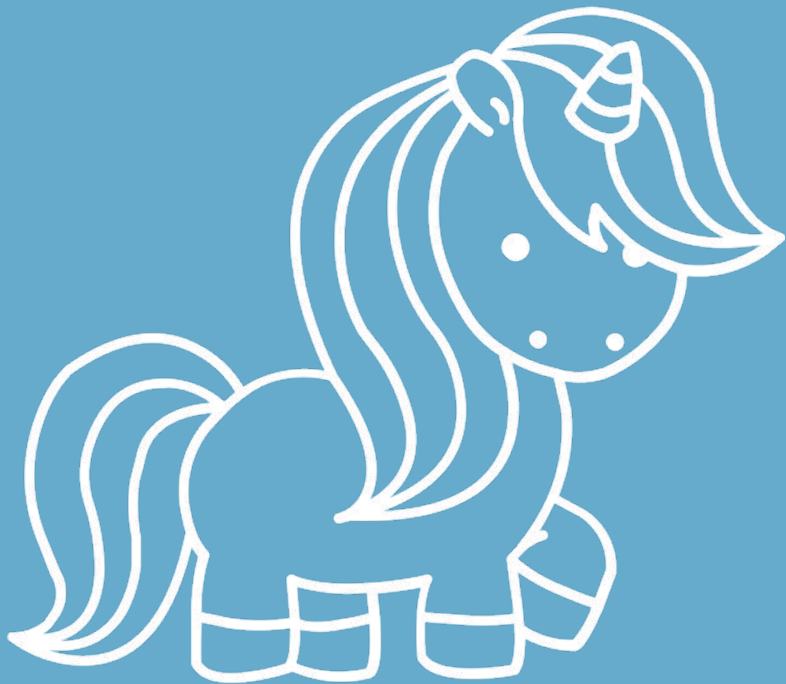


- Sources diverses (API navigateur, vidéo, utilisateurs·ices)
- Indicateurs de poids, de quantité, jalons temporels...
- Indicateurs d'affichage, d'interactivité réelle ou supposée
- Indicateurs techniques ou créés par le métier

C'est la jungle !



Comprendre
Collecter & Automatiser
Savoir-faire, faire savoir
Maintenir



« Temps de chargement »



Les « temps serveur »

- côté serveur, avec une éventuelle instrumentation (Blackfire, New Relic) ;
- côté client : **Time To First Byte (TTFB)**
 - ⚠ bande passante
 - ⚠ latence
 - ⚠ proxys (CDN)



Les évènements navigateur

- **DOMContentLoaded**

aka `performance.timing.domContentLoadedEventEnd`

- **load** ou **onload**

aka `performance.timing.loadEventEnd`

[Performance API sur MDN](#)



Les « visuels »

- **Start Render / First Paint** : la page n'est plus « blanche »

Via la vidéo ou `performance.getEntriesByName('first-paint')`

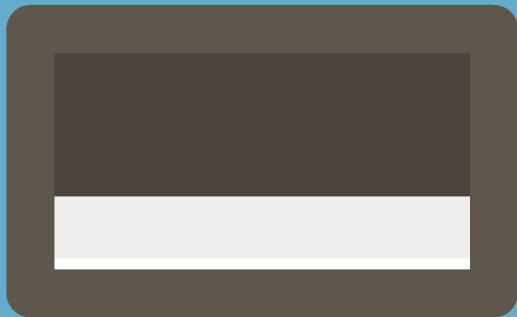
- **First Contentful Paint** : premier pixel non blanc d'une image ou d'un texte.

`performance.getEntriesByName('first-contentful-paint')`

- **Visually Complete** : la partie visible de la page (*viewport*) est entièrement rendue

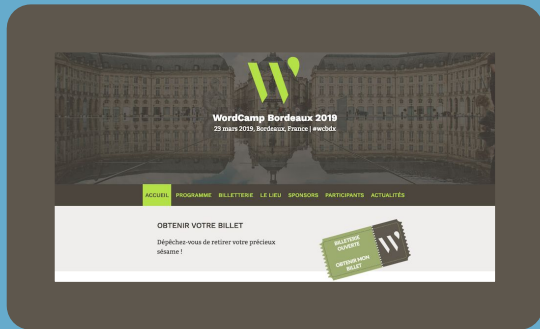


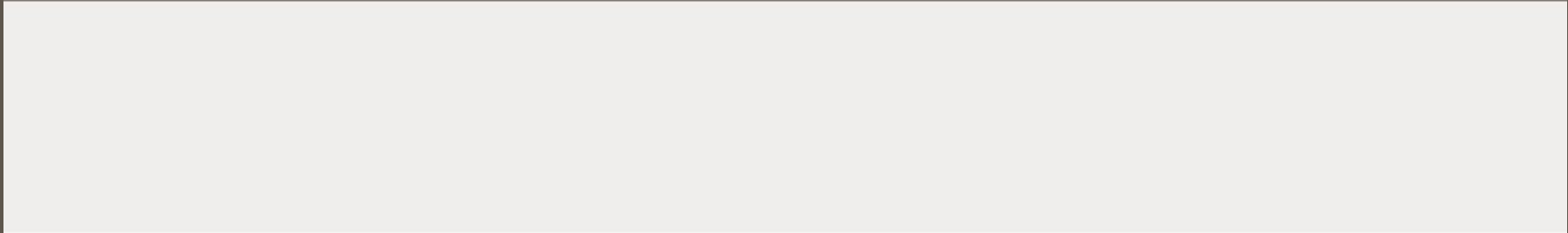
Start Render / First Paint



???

Visually Complete





WordCamp Bordeaux 2019

23 mars 2019, Bordeaux, France | #wcbdx

[ACCUEIL](#)

[PROGRAMME](#)

[BILLETTERIE](#)

[LE LIEU](#)

[SPONSORS](#)

[PARTICIPANTS](#)

[ACTUALITÉS](#)

OBTENIR VOTRE BILLET

Dépêchez-vous de retirer votre précieux
sésame !



WordCamp Bordeaux 2019

23 mars 2019, Bordeaux, France | #wcbdx

ACCUEIL

[PROGRAMME](#)

[BILLETTERIE](#)

[LE LIEU](#)

[SPONSORS](#)

[PARTICIPANTS](#)

[ACTUALITÉS](#)

OBTENIR VOTRE BILLET

Dépêchez-vous de retirer votre précieux
sésame !



WordCamp Bordeaux 2019

23 mars 2019, Bordeaux, France | #wcbdx

ACCUEIL

PROGRAMME

BILLETTERIE

LE LIEU

SPONSORS

PARTICIPANTS

ACTUALITÉS

OBTENIR VOTRE BILLET

Dépêchez-vous de retirer votre précieux
sésame !





WordCamp Bordeaux 2019

23 mars 2019, Bordeaux, France | #wcbdx

ACCUEIL

[PROGRAMME](#)

[BILLETTERIE](#)

[LE LIEU](#)

[SPONSORS](#)

[PARTICIPANTS](#)

[ACTUALITÉS](#)

OBTENIR VOTRE BILLET

Dépêchez-vous de retirer votre précieux
sésame !



Speed Index par l'exemple



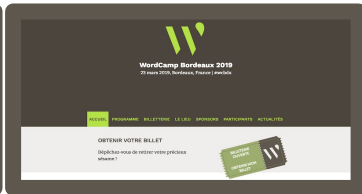
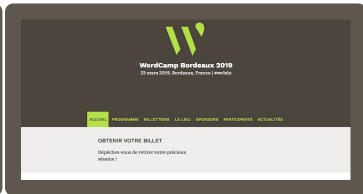
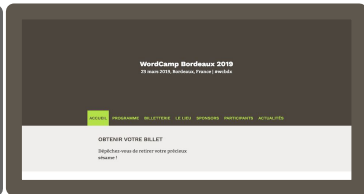
200 ms

400 ms

600 ms

800 ms

1000 ms



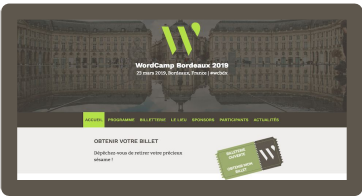
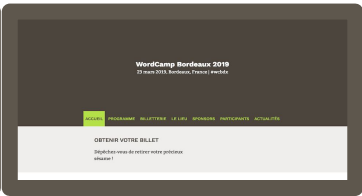
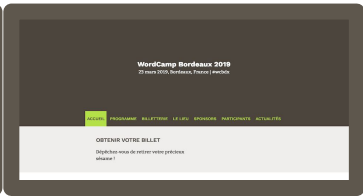
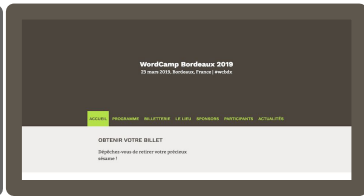
0 %

60 %

75 %

90 %

100 %



0 %

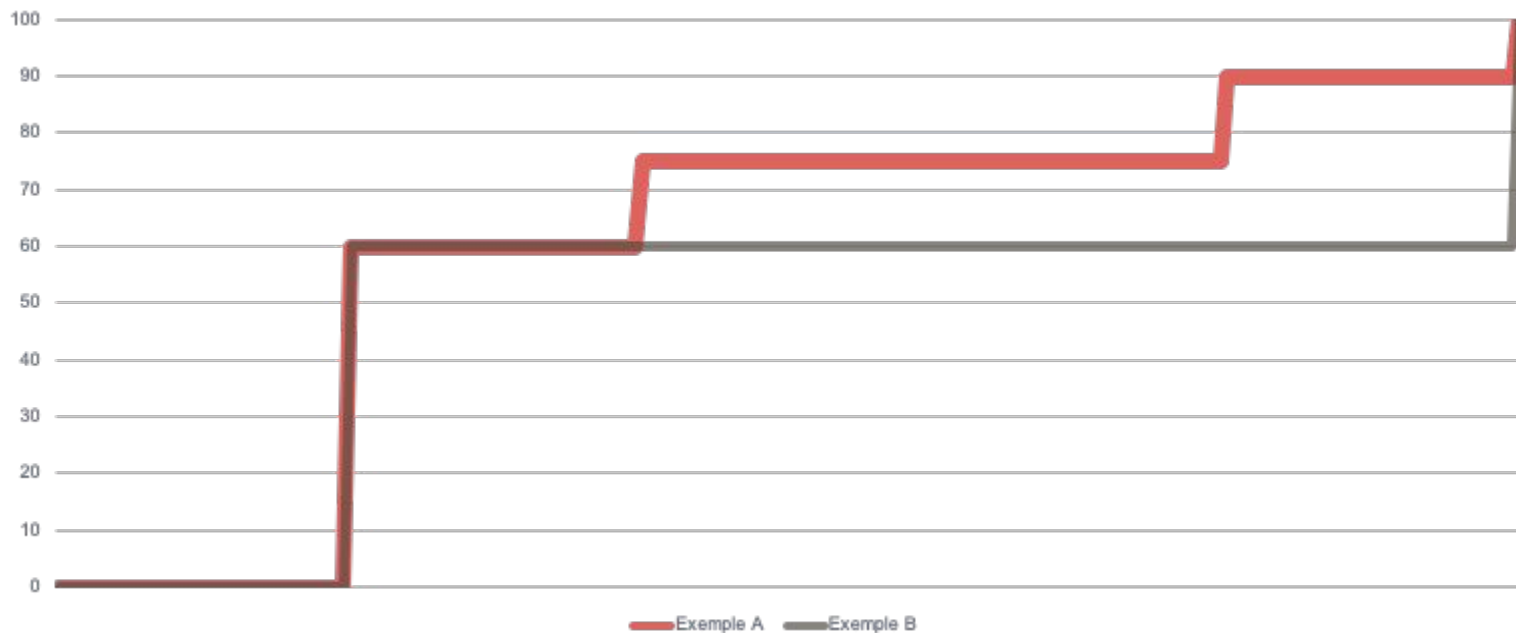
60 %

60 %

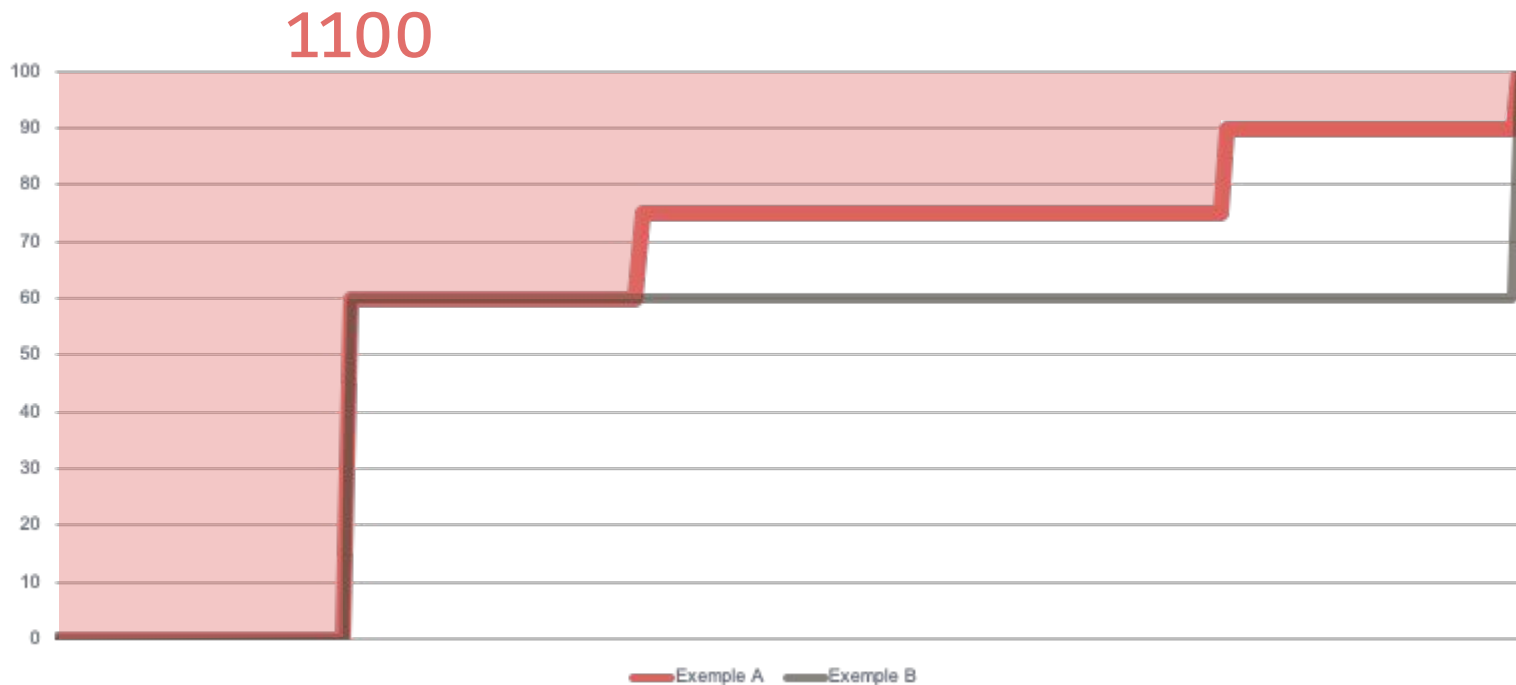
60 %

100 %

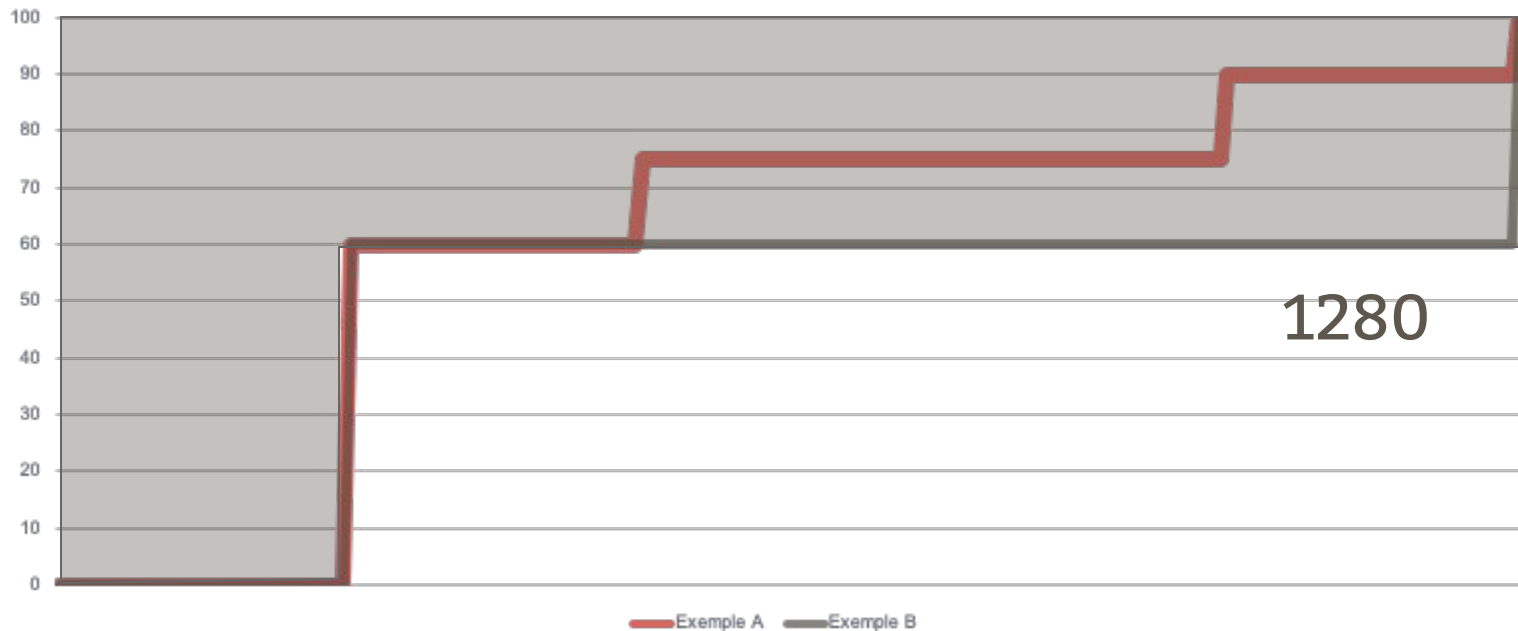
Speed Index par l'exemple



Speed Index par l'exemple



Speed Index par l'exemple



Speed Index par l'exemple



1100



1280

Speed Index



- Mesure la **progressivité** de l'affichage par le calcul du nombre de pixels de chaque couleur.
- Un bon indicateur de l'UX (eXpérience Utilisateur)
- Un indicateur pas si simple, perturbé par :
 - Les fonds colorés (état final + éloigné de l'état initial)
 - Tout ce qui bouge : carrousels, modales, vidéos
 - Les mauvais encodages vidéos
 - Les répartitions des pixels de couleurs =

L'interactivité réelle



- **Delayed Interactions** : à quelle fréquence l'interaction de l'utilisateur a-t-elle été retardée de plus de 50 ms ? Se base sur le **First Input Delay**.
- **Rage Clicks** : clics répétés et rapides sur une zone du *viewport*
 - informe sur l'agacement des utilisateurs
 - entre 1,25 et 1,5 fois le temps nécessaire à l'affichage*

L'interactivité réelle



Mouse Movements / Cursor Trashing : enregistrement des mouvements rapides de souris sur l'interface

- informe(ra) sur l'efficacité de l'interface

*+ d'infos : ["User Experience & Performance: Metrics that Matter", Philip Tellis](#)

Time To Interactive



- Ne mesure pas l'interactivité mais les conditions d'une « **interaction qualitativement garanties** »
- Celui de Google se base sur des fenêtres d'évaluation de 5s
 - + les Long Tasks JavaScript (50 ms)
 - + le trafic réseau
 - + d'autres événements

*+ d'infos : ["Measuring Interactivity with TTI: Time To \(consistently\) Interactive"](#)



Synthetic Real User (JS) Chrome UX Report



Mes premières données



- **Les mesures empiriques**, c'est bien, mais ça ne donne pas une information de qualité.
- **Données ponctuelles** : PageSpeed Insights, Dareboost
 - Vision objective, limitée au contexte et au moment
 - Conseils et bonnes pratiques


Google Analytics



- « Document Content Loaded Time » ou un Custom Timing
 - Éviter l'indicateur de « Page Load Time »
- Échantillonner à 100% en dessous de 20k PV/jour
(siteSiteSampleRate)
- Segmenter pour analyser : matériel, navigateur, FAI...

Surveillance synthétique



- Contextes limités mais maîtrisés
 - Suivi dans le temps
 - Validation d'hypothèses
 - Pas d'installation (permet la mesure de concurrents)
- Stabilité permettant la levée d'alertes
- Parcours utilisateurs aussi
-  **dareboost** mais pas uniquement



Real User Monitoring ?

- Coûts d'implémentation (dév.) et d'exploitation (PV)
- Difficile de déterminer des pistes d'optimisation
- Impossible de tester en dehors de la Production
- MAIS très bon pour le suivi de la performance réelle

⇒ souvent une pratique d'entreprise mûre sur la WebPerf



Comprendre
Collecter & Automatiser
Savoir-faire, faire savoir
Maintenir

Bonnes pratiques WebPerf



- Testables
- Consensuelles
- Pas toujours universelles, mais presque
(nécessité d'analyser les technologies employées)
- Certaines de l'ordre de la configuration
- D'autres sont des *features* à part entière



Les grands thèmes

- **Optimisation de la diffusion** : config serveur(s), protocole(s), CDN(s)
- **Optimisation du rendu** : ressources critiques, ressources d'amélioration
- **Réduction des délai d'interaction**

Quelques indémoudables



- Un code **HTML, CSS, et JS** taillés avec soin, valide et épuré
 - CSS Purging ; JS tree-shaking... automatisez !
- Des domaines spécialisés (pages, assets)
des web serveurs adaptés, et du cache (Varnish, CDN...)
- Des polices de caractères optimisées (subset) et non-bloquantes pour le rendu (utilisez `font-display`)



Généralités serveurs

- Réduire le nombre de requêtes BDD
- Du *Caching* partout où c'est possible
 - Cache objet
 - Cache de pages
 - Transformation en pages statiques



Images

- adaptées à la surface de rendu et aux qualités de l'écran
(srcset, picture)
- optimisées : le bon format en fonction du contenu,
optimisé suivant le navigateur (GIF animés => video)
- chargées uniquement si présentes dans le viewport
(lazy loading)

Images : services dédiés



[https://res.cloudinary.com/borisschapira/image/fetch/e_blur:200,c_crop,ar_1200:600,b_white/e_grayscale/w_1200/b_rgb:0a69c7,o_20/w_1000,c_fit,l_text:PT%20Sans_83:Mesurer%20l%E2%80%99interactivit%C3%A9%20avec%20le%20TTI%20%E2%80%93%20Time%20To%20\(consistently\)%20Interactive,x_2,y_-68,co_black,o_80/w_1000,c_fit,l_text:PT%20Sans_83:Mesurer%20l%E2%80%99interactivit%C3%A9%20avec%20le%20TTI%20%E2%80%93%20Time%20To%20\(consistently\)%20Interactive,y_-70,co_white/l_text:PT%20Sans_50:borisschapira,g_south_east,x_64,y_55,co_black,o_20/l_text:PT%20Sans_50:borisschapira,g_south_east,x_66,y_57,co_white/c_fill,g_south_east,r_max,h_45,l_twitter,w_45,x_356,y_60/c_scale,g_south_west,l_dareboost-logo,w_150,x_60,y_40/https://boris.schapira.dev/assets/images/2019-05-16/carnival.jpg](https://res.cloudinary.com/borisschapira/image/fetch/e_blur:200,c_crop,ar_1200:600,b_white/e_grayscale/w_1200/b_rgb:0a69c7,o_20/w_1000,c_fit,l_text:PT%20Sans_83:Mesurer%20l%E2%80%99interactivit%C3%A9%20avec%20le%20TTI%20%E2%80%93%20Time%20To%20(consistently)%20Interactive,x_2,y_-68,co_black,o_80/w_1000,c_fit,l_text:PT%20Sans_83:Mesurer%20l%E2%80%99interactivit%C3%A9%20avec%20le%20TTI%20%E2%80%93%20Time%20To%20(consistently)%20Interactive,y_-70,co_white/l_text:PT%20Sans_50:borisschapira,g_south_east,x_64,y_55,co_black,o_20/l_text:PT%20Sans_50:borisschapira,g_south_east,x_66,y_57,co_white/c_fill,g_south_east,r_max,h_45,l_twitter,w_45,x_356,y_60/c_scale,g_south_west,l_dareboost-logo,w_150,x_60,y_40/https://boris.schapira.dev/assets/images/2019-05-16/carnival.jpg)

Mesurer l'interactivité avec le TTI – Time To (consistently) Interactive



borisschapira



Et niveau JavaScript ?

Se reposer, autant que possible, sur HTML et CSS, voire n'utiliser JS que pour de l'amélioration progressive

- Réduire le JS au maximum (faire en CSS tout ce qui est faisable)
- Différer le JS restant ou l'injecter à la demande



Comment faire le tri ?

- Garder l'historique (git, SVN, noeuds statiques)
- Itérer, tester, communiquer
 - Audits synthétiques : comparaison de **versions A / B**
 - Des tests fonctionnels, pour éviter **les régressions**
 - Envisager des procédures de désactivation des optims



Comprendre
Collecter & Automatiser
Savoir-faire, faire savoir
Maintenir



Une fois en Production

- On continue de mesurer pour vérifier qu'on respecte ses [budgets de performance](#)
- On éviter les optimisations à la volée et les montées de versions de plugins

(vous étiez déjà au courant, HEIN, on est d'accord ?!)

Réviser la chaine de Prod

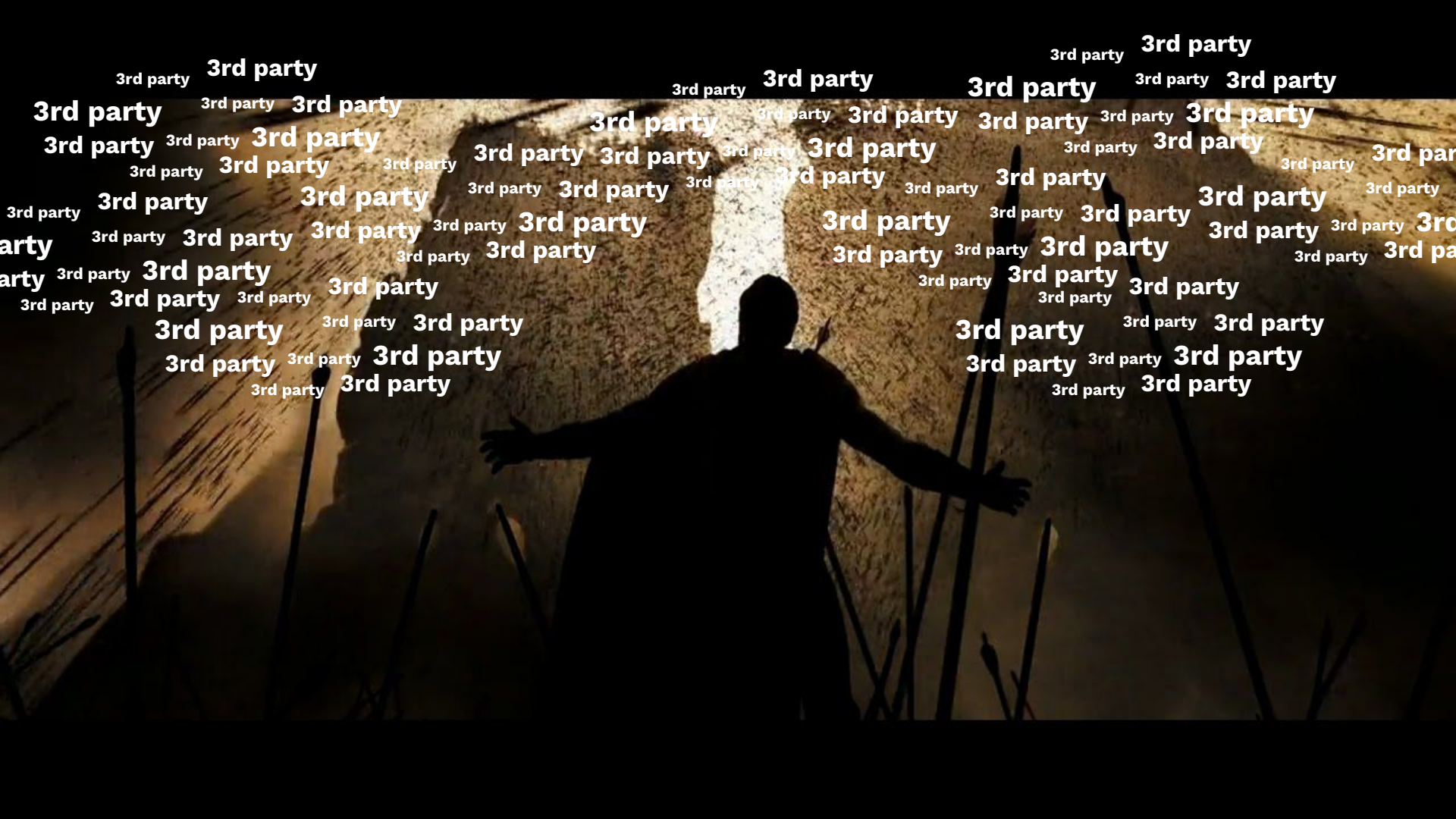


- La WebPerf doit être, comme l'accessibilité ou le référencement, **une valeur de premier ordre**
- Fonctionnellement, beaucoup de concepts ne verraient pas le jour si c'était le cas : identifier les problèmes au plus tôt en instaurant **une culture** de la web performance
- Adoptez une vision concurrentielle pour **éviter la surqualité**

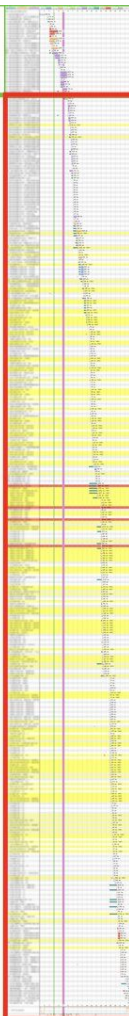
Et après c'est fini ?

Non, après on commence la vraie lutte...





This is your website...



...this is your website
on tag managers.



Merci !

 **dareboost**
testez !

we  speed
proposez, venez !