

# Surveiller · Optimiser **Performance Web**



<https://boris.schapira.dev/psconnect-bayonne>



**Boris  
Schapira**



we  speed

**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**



# Une demande en augmentation

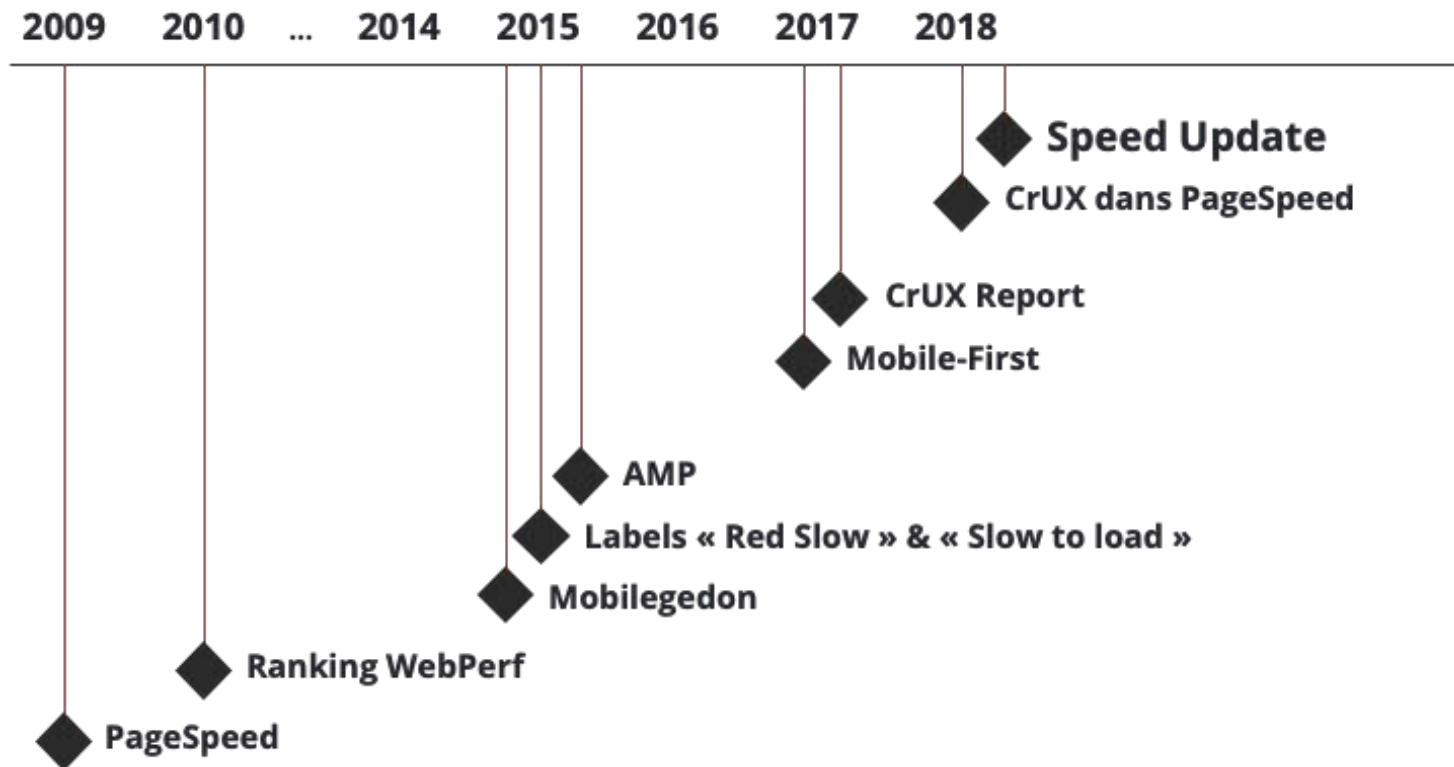
- Expérience *numériques* ++
- Multiplication des canaux (RS)
- Gains technologiques (ex: 5G)
- Concurrence des plate-formes (AMP)
- **Mais**, effet rebond qui nécessite un écoconception

# Alors la technique s'améliore

- Maîtrise des mises-en-page adaptatives
- Formats et tailles d'image suivant les besoins
- Optimisation dans la diffusion des contenus (CDN)
- Nouvelles architectures (microservices, serverless)
- Déconstruction du concept de « page »



# Google, WebPerf 2009-2018



# Impacts

-  Coût d'exploitation (Réseau, CPU)
- UX++  Engagement & Conversion
- Augmentation des gains publicitaires
- Intégration des problématiques de Qualité Web à la culture de l'entreprise

# Premières analyses “one-shot”

## Analyse de qualité et performance web

<https://www.albertine-swim.com/fr/>

Rapport privé généré le 3 juil. 2019 18:34:40 | [Relancer](#)

Télécharger le rapport



Pas mal !  
Mais peut mieux faire.



Problèmes



Améliorations



Succès



Voir les priorités

VISITEUR SIMULÉ : Chrome Paris 8.0/1.5Mbps (Latence : 50 ms) ?

Modifier

Mémoriser

Requêtes



HTML CSS Scripts Images Autres

Poids



Premier octet



Début affichage



Fin chargement



Timeline / Waterfall

Détail des étapes

Le chargement en Images

Erreurs navigateur 0 OK

Support d'HTTP/2 : 89%

Speed Index: 1835

Technologies :



## Bonnes pratiques et conseils : [Toutes les catégories](#)

### Ce que vous pouvez améliorer

Volume de données

0/100

#### 2 images sont redimensionnées côté navigateur

Une page récupérant une image plus grande que celle réellement affichée charge inutilement un volume de données trop important.

Sécurité

0/100

#### Il manque une politique de sécurité sur la provenance de vos ressources

Protégez votre site Web contre les attaques de type XSS (Cross-Site Scripting) en mettant en place une politique restrictive de sécurité du contenu.

Optimisation du rendu

60/100

#### Différez l'utilisation du code JavaScript

Lorsque le navigateur web rencontre du code JavaScript en interprétant le code source d'une page web, cela peut ralentir considérablement l'affichage de la page, surtout s'il est nécessaire de télécharger un script externe.

Sécurité

0/100

#### 17 cookies ne sont pas sécurisés

Un cookie transmis au navigateur via le protocole HTTPS doit pouvoir transiter uniquement sur une connexion sécurisée (sauf cas particulier).

Référencement (SEO)

0/100

#### Cette page n'indique pas de fil d'Ariane aux moteurs de recherche

Vous pouvez mettre en valeur vos pages sur les moteurs de recherche en indiquant un fil d'Ariane via des données structurées.

# Premières analyses “one-shot”

Analyse de qualité et performance web

<https://www.albertine-swim.com/fr/>

Rapport privé généré le 3 juil. 2019 18:34:40 | Relancer

Télécharger le rapport



Pas mal !  
Mais peut mieux faire.

3

Problèmes

4

Améliorations

85

Succès



Voir les priorités ▼

VISITEUR SIMULÉ : Chrome Paris 8.0/1.5Mbps (Latence : 50 ms) ?

Modifier

Mémoriser

Requêtes



HTML CSS Scripts Images Autres

Poids



Premier octet



Début affichage



Fin chargement



Timeline / Waterfall

Détail des étapes

Le chargement en Images

Erreurs navigateur 0 OK

Support d'HTTP/2 : 89%

Speed Index: 1835

Technologies :



**Indicateurs  
(nombres,  
poids,  
jalons  
temporels)**

# Premières analyses “one-shot”

## Analyse de qualité et performance web

<https://www.albertine-swim.com/fr/>

Rapport privé généré le 3 juil. 2019 18:34:40 | Relancer

Télécharger le rapport



Pas mal !  
Mais peut mieux faire.

3

Problèmes

4

Améliorations

85

Succès



Voir les priorités

Conseils  
(bonnes pratiques)

## Bonnes pratiques et conseils : [Toutes les catégories](#)

### Ce que vous pouvez améliorer

Volume de données

0/100

#### 2 images sont redimensionnées côté navigateur

Une page récupérant une image plus grande que celle réellement affichée charge inutilement un volume de données trop important.

Sécurité

0/100

#### Il manque une politique de sécurité sur la provenance de vos ressources

Protégez votre site Web contre les attaques de type XSS (Cross-Site Scripting) en mettant en place une politique restrictive de sécurité du contenu.

Optimisation du rendu

60/100

#### Différez l'utilisation du code JavaScript

Lorsque le navigateur web rencontre du code JavaScript en interprétant le code source d'une page web, cela peut ralentir considérablement l'affichage de la page, surtout s'il est nécessaire de télécharger un script externe.

Sécurité

0/100

#### 17 cookies ne sont pas sécurisés

Un cookie transmis au navigateur via le protocole HTTPS doit pouvoir transiter uniquement sur une connexion sécurisée (sauf cas particulier).

Référencement (SEO)


0/100

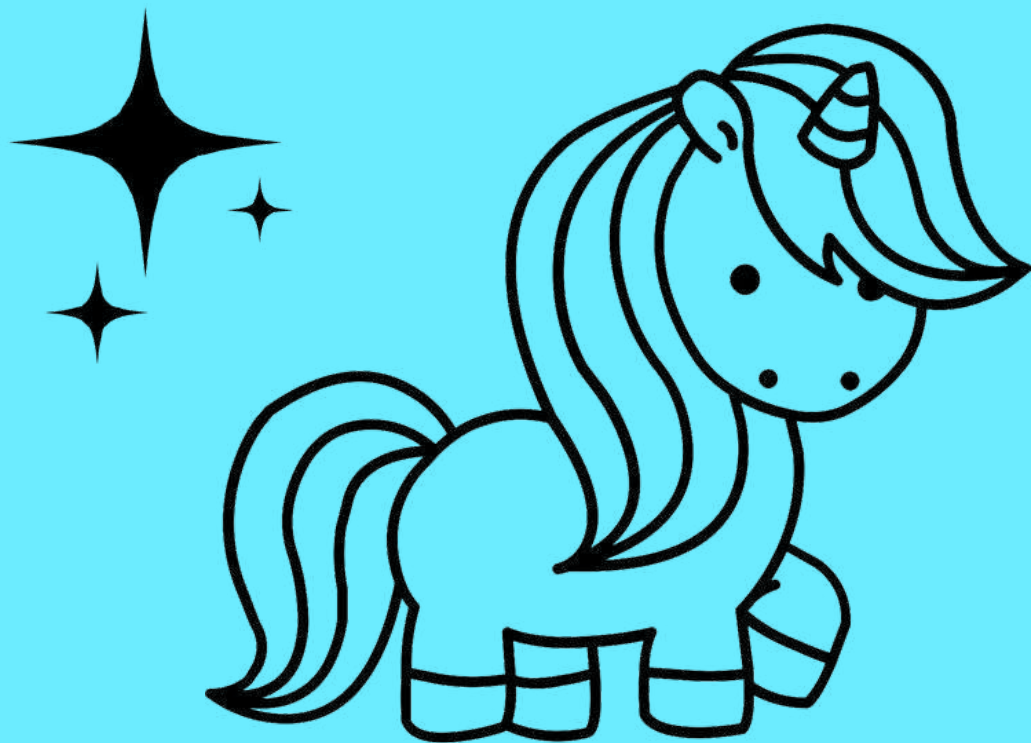
#### Cette page n'indique pas de fil d'Ariane aux moteurs de recherche

Vous pouvez mettre en valeur vos pages sur les moteurs de recherche en indiquant un fil d'Ariane via des données structurées.

**Comprendre**  
**Collecter & Automatiser**  
**Savoir-faire, faire savoir**  
**Maintenir**

# 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



**« Temps de chargement »**



# 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**

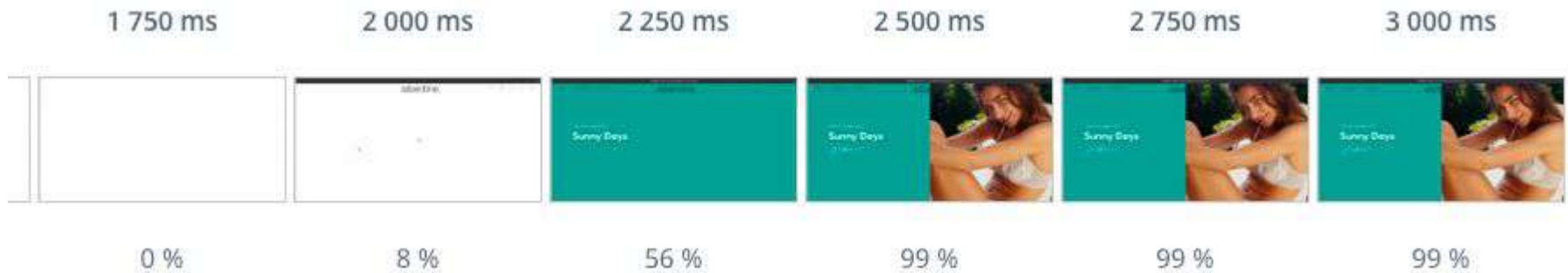
# Les « temps serveur »

- côté serveur, avec une éventuelle instrumentation ([Blackfire.io](https://blackfire.io), par exemple)
- côté client : **Time To First Byte (TTFB)**



# Les « visuels »

- **Start Render / First Paint** : la page n'est plus « blanche »
- **Visually Complete** : la partie visible de la page (*viewport*) est entièrement rendue
- **et entre les deux ?**



# Speed Index

- Mesure la **progressivité** de l'affichage
- Attention, perturbé par :
  - Les pop-ins (promotions, *EU Cookie Law*)
  - Les carrousels, modales, vidéos...



- 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

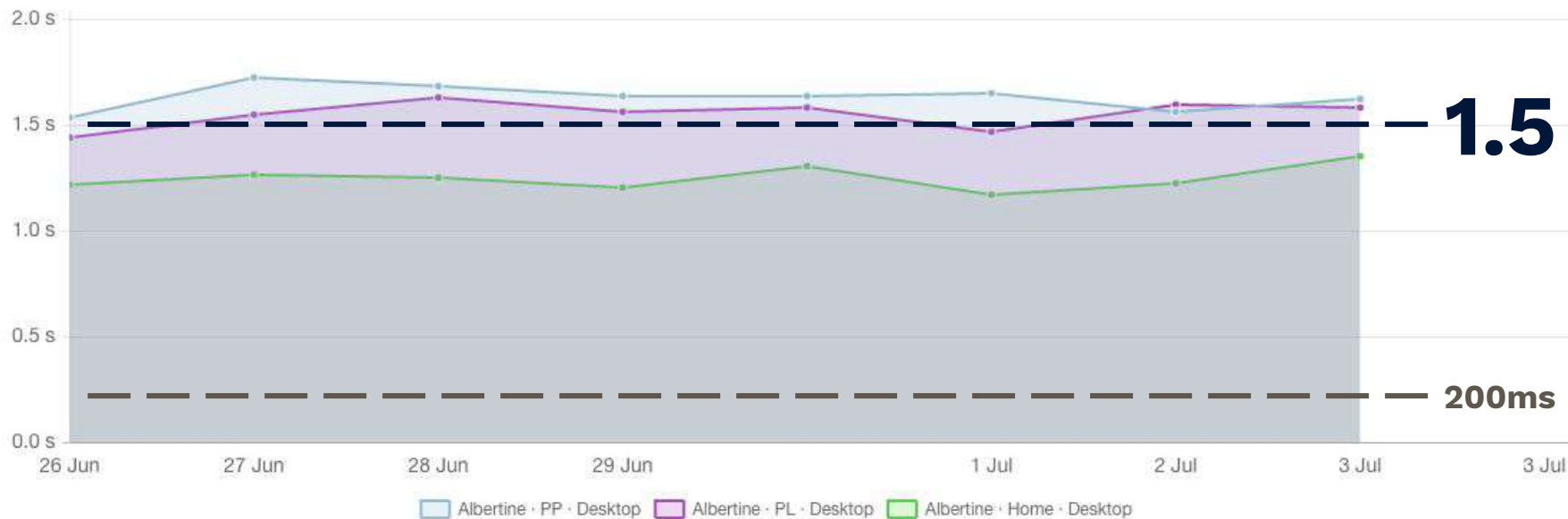
\*+ d'infos : ["Measuring Interactivity with TTI: Time To \(consistently\) Interactive"](#)

# Albertine · Premier Octet

Chrome 73 | Paris | 80 percentiles, agrégation quotidienne

PSGCONNECT  
BAYONNE

🕒 Premier octet ▾



— 1.5 s

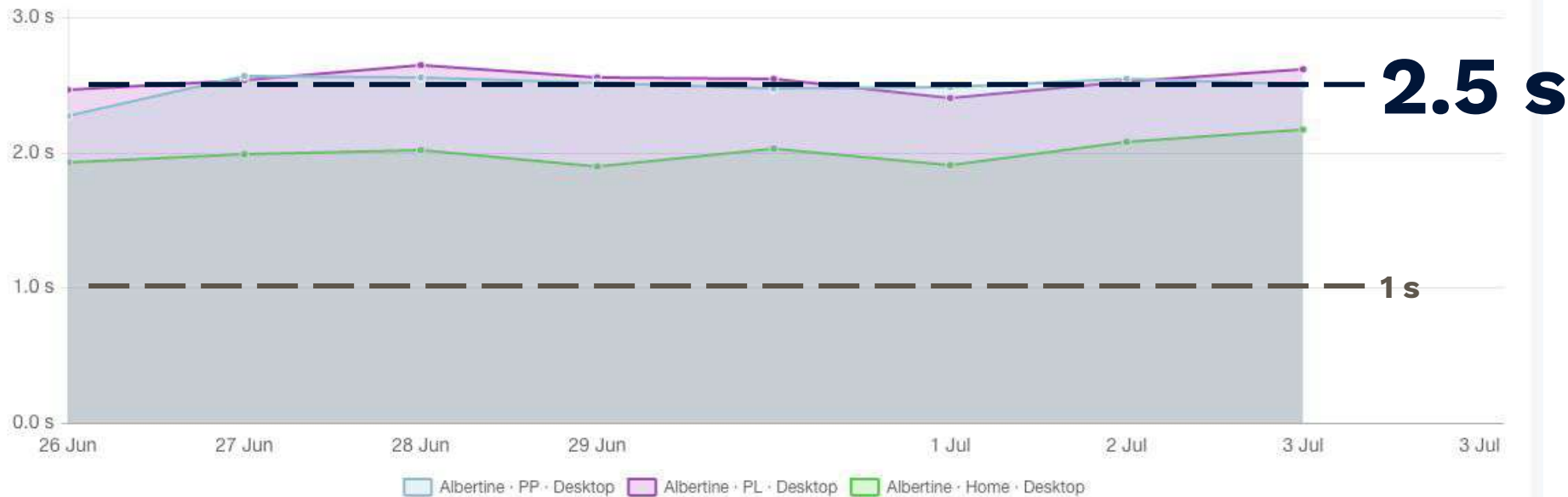
— 200ms

# Albertine · Speed Index

Chrome 73 | Paris | 80 percentiles, agrégation quotidienne

PSGCONNECT  
BAYONNE

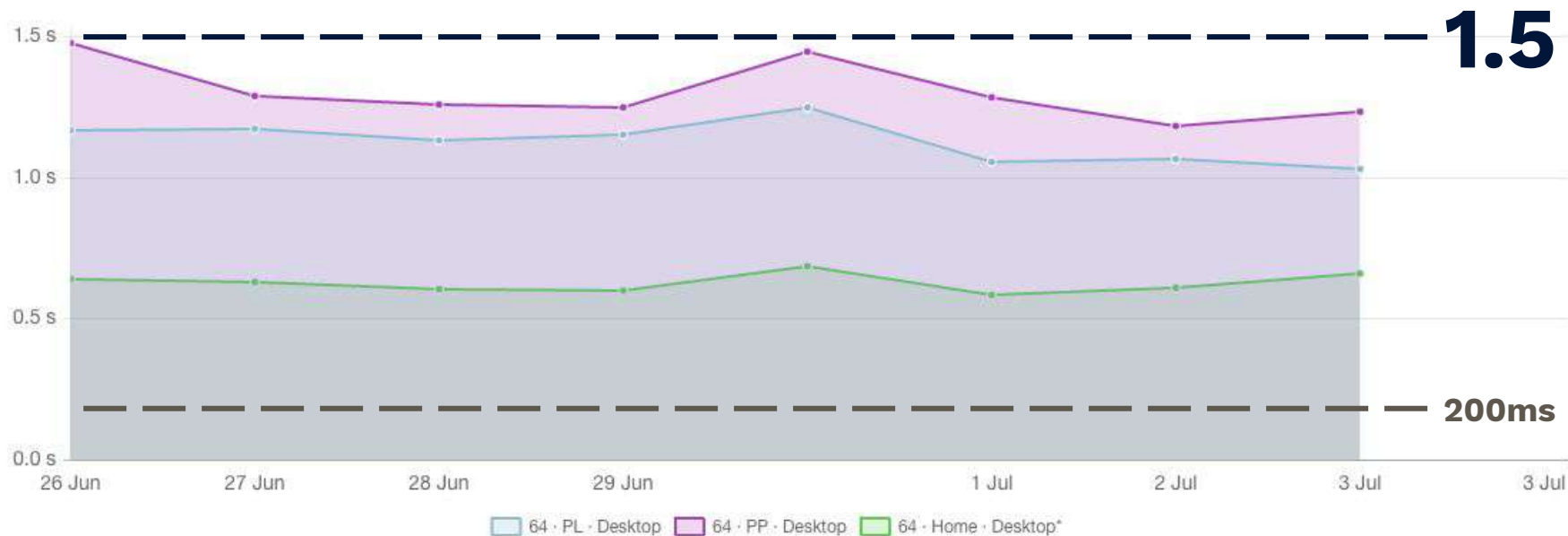
Speed Index ▾



# 64 • Premier Octet

Chrome 73 | Paris | 80 percentiles, agrégation quotidienne

🕒 Premier octet ▾



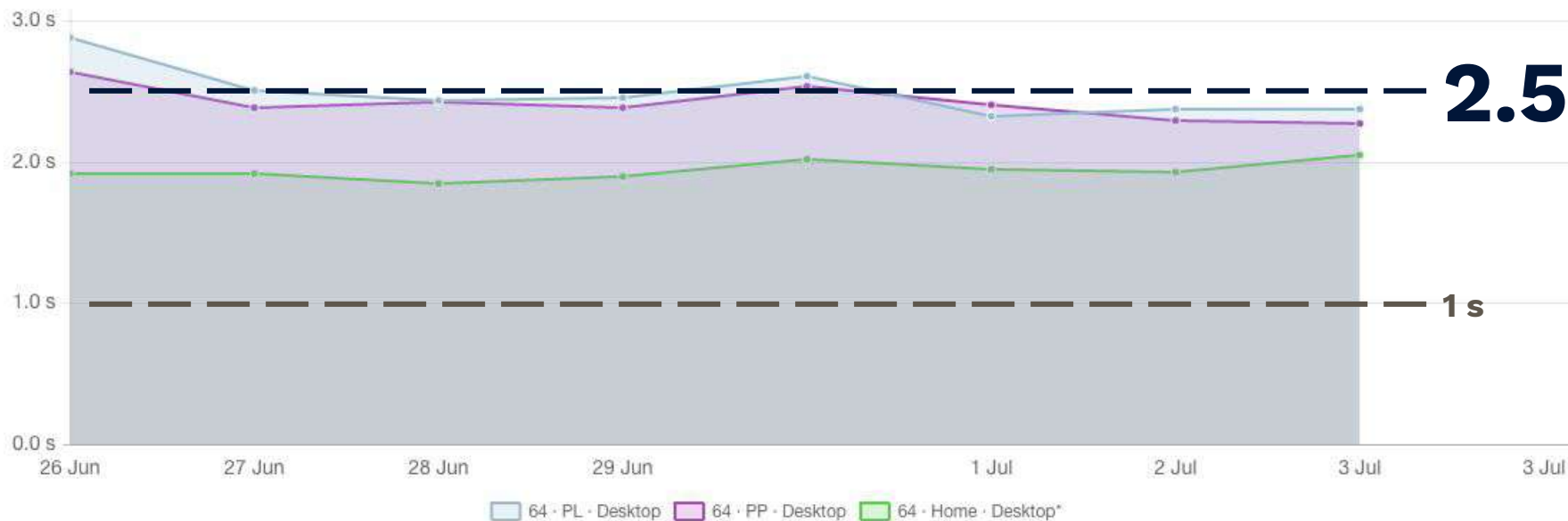


# 64 · Speed Index

Chrome 73 | Paris | 80 percentiles, agrégation quotidienne

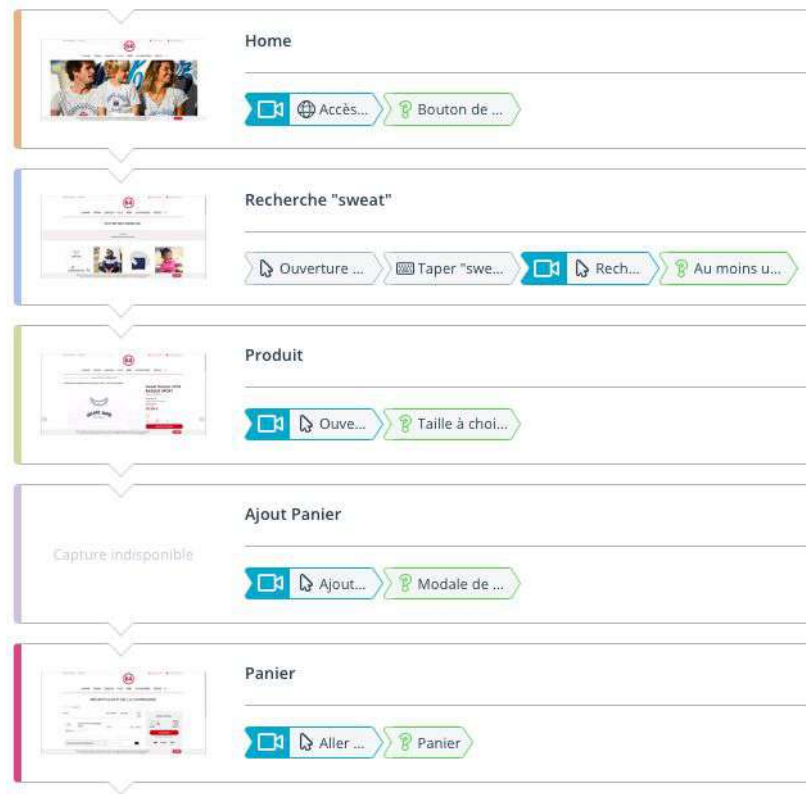
PSGCONNECT  
BAYONNE

🕒 Speed Index ▾



# Surveiller son tunnel

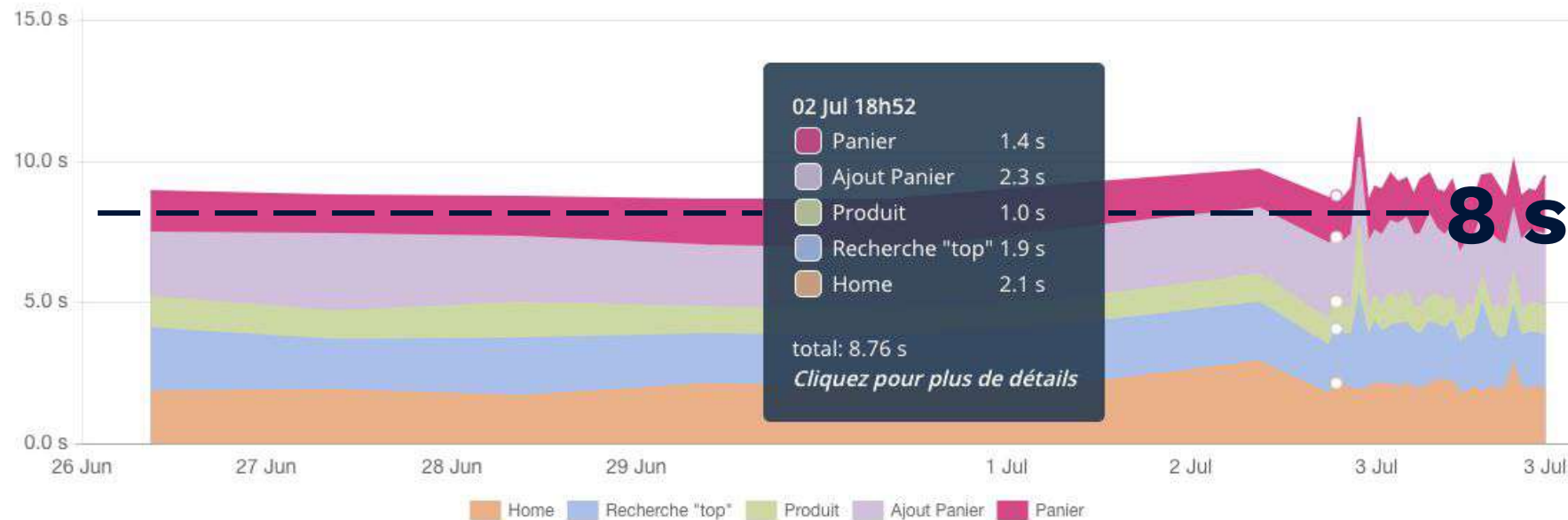
L'expérience utilisateur n'est pas composée de chargement de pages mais d'(inter)actions, avec une notion d'étapes, et de cache navigateur.



# Albertine · Speed Index cumulé

Chrome 73 | Paris | 80 percentiles, agrégation quotidienne

🕒 Speed Index ▾



# 64 • Speed Index cumulé

Chrome 73 | Paris | 80 percentiles, agrégation quotidienne

🕒 [Speed Index](#) ▾



# RUM : Real User Monitoring

- Coûts d'implémentation (dév.) et d'exploitation ++
- 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

- **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\*

# RUM : 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](#)

**Comprendre**  
**Collecter & Automatiser**  
**Savoir-faire, faire savoir**  
**Maintenir**



# Corpus de “Bonnes pratiques”

- Consensuelles
- Réalistes
- Vérifiables (automatiquement)
- Dépendantes des technologies employées

Technologies :



# Des conseils détaillés pour monter en compétence

## ! 2 images sont redimensionnées côté navigateur

Une page récupérant une image plus grande que celle réellement affichée charge inutilement un volume de données trop important.

### Redimensionnement d'images

Le redimensionnement d'une image côté navigateur pour en réduire la taille d'affichage n'est pas recommandé.

Fournissez directement vos images aux dimensions d'affichage utilisées sur votre site. Vous évitez ainsi l'envoi de données inutiles sur le réseau, ce qui diminue le temps de chargement de la page.

### Responsive design ? Images Retina ?

Le responsive design ou encore l'utilisation d'images retina ne justifient pas un redimensionnement d'images. Même dans ces cas, différentes méthodes existent afin de délivrer vos images aux bonnes tailles. Nous vous conseillons de lire les ressources suivantes :

- [Introduction sur les images responsive](#)
- [Picturefill, pour supporter l'élément <picture> dès maintenant \(EN\)](#)
- [RICG, groupe de travail sur les images responsive \(EN\)](#)

Ne redimensionnez pas les images suivantes :

- [www.albertine-swim.com/modul\[...\]nserie.jpg](http://www.albertine-swim.com/modul[...]nserie.jpg) (taille d'affichage : 676x688)
- [www.albertine-swim.com/modul\[...\]tropic.jpg](http://www.albertine-swim.com/modul[...]tropic.jpg) (taille d'affichage : 340x510)

# Ne pas tomber dans le piège

- Certaines optimisations sont simples à comprendre et difficiles à implémenter. Pour d'autres, c'est l'inverse.
- Mon astuce :
  - Est-ce une configuration ?  
⇒ **Probablement complexe.**
  - Est-ce une fonctionnalité à part entière ?  
⇒ **Probablement compliqué.**



# Les grands thèmes

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

# Quelques indémodables

- 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`)

- Réduire le nombre de requêtes BDD
- Du *Caching* partout où c'est possible
  - Vérifiez que le système de cache Smarty est bien activé via l'interface d'administration de votre Prestashop (Préférences > performances).  
Le **mode debug** peut vous aider.

- **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)

# 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
  - Anticiper la désactivation des optimisations
  - Avoir des outils permettant de communiquer transversalement sur les résultats

**Comprendre**  
**Collecter & Automatiser**  
**Savoir-faire, faire savoir**  
**Maintenir et dépasser**

# Une fois en Production

- On contrôle ses budgets de performance



# Réviser la chaîne de Production

- 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**



Merci !

 **dareboost**  
testez !

w e  s p e e d  
v e n e z !