RAPPORT · 01M5FBQC2STEN72BYDKQJPXS6A

https://teelda.com/

audite le 21/10/2026 a 13:52 · 47 criteres analyses · duree 5,7s

▸ VERDICT · SITE CORPORATE / LANDING · CONFIANCE 50 %

Site *equilibre* · *bien optimise* techniquement.

Volume maitrise et pratiques techniques solides. Quelques gains possibles sur les tiers et la compression des images.

01 · EMPREINTE CO2

B

0,068 g / visite · 8,19 kg / an

Empreinte correcte, proche de la mediane des sites audites. Des gains restent possibles sur le volume.

02 · ECO-CONCEPTION

A

79 / 100 · 5 familles · volume exclu

Tres bon niveau d'optimisation technique. Quelques reglages fins peuvent encore ameliorer le socle.

03 · CLASSEMENT

53%

des sites corporate sont plus legers que le votre

Votre site est legerement plus lourd que la mediane de la categorie (HTTP Archive 2024, 16 M sites).

04 · REDUCTION

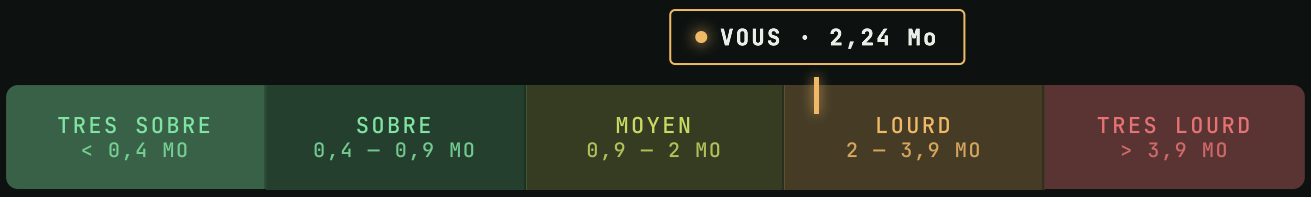
-40%

estimation cumulee des leviers actionnables · 13 recos

Ordre de grandeur calcule en sommant les gains potentiels des recommandations actionnables (hors recos macro et hors recos sans gain octets), avec une fourchette d'incertitude +/- 50 % cumulee a celle du modele CO2. Plafonne dynamiquement selon la lettre Empreinte du site : A+ 15 %, A 25 %, B 40 %, C 55 %, D 70 %, E/F 80 %. Justification : un site deja sobre a moins de marge structurelle.

▸ POIDS TRANSFERE PAR VISITE · VOTRE SITE VS. SITES CORPORATE

Votre page pese 2,24 Mo. Mediane corporate ≈ 2,00 Mo.



/EMPREINTE CARBONE

Emissions estimees

PAYS D'AUDIENCE

France | Europe | Monde

VISITES / MOIS

1 k | 10 k | 100 k | 1 M

PAR VISITE

0,068 gCO2e

fourchette [0,005 - 0,101] selon coefficient energetique (SWD v4 / Aslan 2018)

PAR AN, BASE 120 000 VISITES

8,19 kgCO2e

extrapolation sur 10 000 visites / mois (modifiable ci-dessus)

KM VOITURE THERMIQUE	KM TGV
42 km	3412 km
RECHARGES SMARTPHONE	HEURES DE STREAMING HD
987	149 h

ARBRES ABSORBANT/AN ?	
0,3	

ⓘ La **lettre de l'empreinte** affichee dans le hero est calculee sur une reference audience mondiale figee (442 gCO2/kWh) pour rester comparable entre sites. Les chiffres ci-dessus refletent l'audience et le trafic selectionnes.

/HEBERGEMENT

Infrastructure

ENERGIE RENOUVELABLE	HEBERGEUR
declare eco-responsable	O2Switch
LOCALISATION SERVEUR	**INTENSITE DATACENTER**
France (FR)	50 gCO2/kWh
REACTIVITE SERVEUR (TTFB)	
38 ms	

▸ **Declaration de l'hebergeur** (non certifie par Green Web Foundation) : O2Switch declare utiliser ~94 % d'energie decarbonnee (2019) et s'engage a ne pas installer ses datacenters dans des zones dependant principalement d'energies fossiles.
Source : https://www.o2switch.fr/green/ (declaration 2019)

▸ intensite carbone appliquee : 50 gCO2/kWh sur la part datacenter (22%), 57 gCO2/kWh sur la part reseau + device (78%, audience **France**).

▸ TTFB (time to first byte) : temps ecoule entre la requete HTTP et le premier octet recu. Indicateur observable de la reactivite serveur + latence reseau. Reperes usuels : < 200 ms (bon), 200-600 ms (acceptable), > 600 ms (lent). Non integre au score CO2 et non interprete automatiquement.

/CRAWL

Ce qui a ete telecharge

POIDS TOTAL	REQUETES	DUREE AUDIT
2.2 Mo	51	5,7s

TROIS MESURES DU POIDS, UNE SEULE SERT A LA LETTRE EMPREINTE CO2

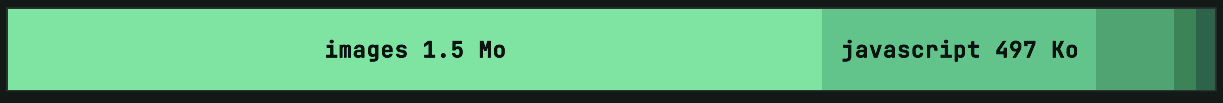
PHASE 1	poids initial (avant scroll, ce que voit un visiteur fond de page)		784 Ko · 25 requetes
PHASE 2	poids complet apres scroll (worst case, brut transfere)	*= KPI « poids total » ci-dessus, comparable a HTTP Archive*	2.2 Mo · 51 requetes
PHASE 3	**poids retenu pour la lettre Empreinte**	(moyenne ponderee, voir formule ci-dessous)	**1.5 Mo**

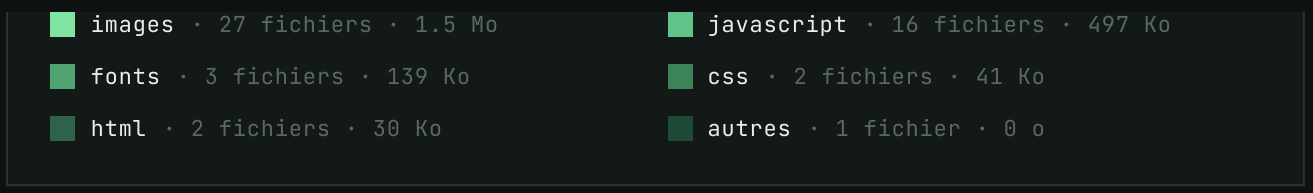
ETAPE 1 (LAZY) : poids effectif = 50 % × 784 Ko (phase 1) + 50 % × 2.2 Mo (phase 2) = 1.5 Mo

ETAPE 2 (CACHE + RETOURS) : cache long detecte = 0 % → **poids retenu = 1.5 Mo** (egal au poids effectif, aucune reduction par cache)

POURQUOI CE POIDS RETENU ?

La lettre Empreinte CO2 reflete ce que vit un visiteur moyen, pas le pire scenario. 50 % des visiteurs ne scrollent pas et ne declenchent pas le lazy loading (etudes UX Nielsen Norman 2024). 25 % des visiteurs reviennent et profitent du cache navigateur (default ajustable par le formulaire). On calcule donc une moyenne ponderee qui recompense les sites qui font du lazy loading et configurent un cache long. Detail methodologique.




images · 27 fichiers · 1.5 Mo
javascript · 16 fichiers · 497 Ko
fonts · 3 fichiers · 139 Ko
css · 2 fichiers · 41 Ko
html · 2 fichiers · 30 Ko
autres · 1 fichier · 0 o

/DETAIL PAR FAMILLE

6 familles d'analyse

A

58 / 100

Volume de donnees

2 point(s) a ameliorer · A1 · A6

B

54 / 100

Efficacite du transfert

6 point(s) a ameliorer · B6 · B4

C

97 / 100

Sobriete medias

aucun point d'attention

D

86 / 100

Tiers et trackers

2 point(s) a ameliorer · D2 · D3

E

100 / 100

Hebergement

1 point(s) a ameliorer · E2

F

72 / 100

Device utilisateur

2 point(s) a ameliorer · F2 · F4

-40 %

Vous pouvez reduire significativement l'empreinte carbone de cette page en appliquant les recommandations ci-dessous.

actuel : **8,19 kg CO2e / an** → cible : **4,91 kg CO2e / an**

Estimation cumulee des leviers actionnables (hors recos « vue d'ensemble » et hors recos sans gain octets), plafonnee selon la lettre Empreinte du site (A+ 15 % / A 25 % / B 40 % / C 55 % / D 70 % / E-F 80 %). Fourchette d'incertitude +/- 50 % qui se cumule a celle du modele CO2. Les gains reels dependent de la mise en oeuvre effective. Base : 10 000 visites / mois.

/RECOMMANDATIONS

13 actions classees par impact

01 ● CRITIQUE

vue d'ensemble
(detail dans les recos suivantes)

Votre page est trop lourde au total

> Votre page pese 2.2 Mo au total. Une page sobre vise 500 Ko a 1 Mo maximum.

POURQUOI C'EST IMPORTANT

Chaque octet telecharge consomme de l'energie - sur le reseau et sur l'appareil du visiteur. Plus une page est lourde, plus son empreinte carbone par visite est elevee. C'est un signal global : le gain reel viendra des recommandations actionnables ci-dessous (formats d'images, compression, JS, tiers, lazy loading) qui detaillent comment alleger la page.

BONNE PRATIQUE

Identifier les ressources les plus lourdes (images, JavaScript, fonts) et les alleger en priorite : compression, formats modernes, suppression des elements non utilises. Viser une page la plus legere possible sans sacrifier l'experience.

source · RGESN 4.1 / GR491 GES-010 / SWD v4 #R1-A1

02 ● CRITIQUE

vue d'ensemble
(detail dans les recos suivantes)

Vos images pesent lourd au total

> Les images representent 1.5 Mo de votre page.

POURQUOI C'EST IMPORTANT

Les images sont souvent le premier poste de poids d'une page. Chaque visite les retelecharge (si absence de cache). Comme A1, c'est un signal global : le gain reel se chiffre via les recommandations actionnables (formats WebP/AVIF, compression, lazy loading, dimensions exactes) qui sont listees plus bas.

BONNE PRATIQUE

Compresser chaque image a la taille reelle d'affichage, utiliser des formats modernes (WebP, AVIF) et mettre en place le chargement differe pour les images situees en bas de page.

source · RGESN 6.9 / GR491 GES-021 #R2-A6

03 ● IMPORTANT

Les images ne s'adaptent pas a la taille de l'ecran

> Ratio d'images avec srcset : 0%.

POURQUOI C'EST IMPORTANT

Sans version adaptative, un telephone telecharge la meme grosse image qu'un ordinateur - plusieurs fois plus qu'il n'en a besoin. Avec un dispositif srcset, le navigateur choisit la taille d'image adaptee a la resolution.

BONNE PRATIQUE

Proposer plusieurs tailles de chaque image (srcset) avec une description des points de rupture (sizes). Les CMS modernes (WordPress, Drupal) le font automatiquement.

source · W3C WSG #R5-B9

04 IMPORTANT

Vos ressources ne sont pas mises en cache longue duree

> Seulement 0 ressource(s) sur 48 ont des entetes de cache superieur a 1 heure.

POURQUOI C'EST IMPORTANT

Sans cache, chaque visiteur retelecharge toutes les ressources a chaque visite. Avec un cache configure, les ressources restent stockees dans le navigateur et ne sont pas redemandees. C'est le plus gros levier sur les visites recurrentes.

BONNE PRATIQUE

Configurer l'entete Cache-Control avec une duree longue (1 an) sur les ressources statiques. Utiliser un nom de fichier versionne (ex: style.v42.css) pour invalider le cache quand vous faites une mise a jour.

source · RGESN 3.5 #R4-B4

05 IMPORTANT

Vos images pourraient utiliser des formats plus modernes

> 12 image(s) au format ancien (JPG, PNG, GIF) contre 15 au format moderne (WebP, AVIF, SVG).

POURQUOI C'EST IMPORTANT

Les formats WebP et AVIF sont concus pour le web : ils offrent la meme qualite visuelle qu'un JPG avec environ 30 a 50% de poids en moins. Tous les navigateurs modernes les supportent.

BONNE PRATIQUE

Convertir les images JPG et PNG en WebP ou AVIF. De nombreux outils font la conversion en un clic. Conserver une version JPG en fallback pour les navigateurs tres anciens via la balise picture.

source · GR491 GES-021 #R3-B6

06 ● MINEUR

Un ou plusieurs outils d'analyse sont actifs

```
1 outil(s) d'analyse : Google Analytics.
```

POURQUOI C'EST IMPORTANT

Les trackers d'analyse (Google Analytics, Hotjar, etc.) injectent du JavaScript qui tourne en continu pendant la navigation. Ils envoient aussi des donnees sur chaque action, ce qui consomme de la bande passante.

BONNE PRATIQUE

Utiliser une alternative plus sobre et respectueuse de la vie privee (Matomo self-hosted, Plausible, Fathom). N'installer qu'un seul outil d'analyse, supprimer les autres.

source · WhoTracks.me #R9-D2

07 ● MINEUR

Certains fichiers JavaScript ne sont pas minifies

```
3 fichier(s) JavaScript sans extension .min.js totalisent 416 Ko. Un JS minifie pese environ 30 a 50 % de moins qu'un JS source.
```

POURQUOI C'EST IMPORTANT

Comme pour le CSS, la minification JavaScript retire les espaces, commentaires et peut meme raccourcir les noms de variables locales (uglification). Sur des gros fichiers, on peut atteindre -50 % de poids sans aucun changement de comportement.

BONNE PRATIQUE

Utiliser un outil de build (Webpack, esbuild, Terser) pour minifier le JS en production. Les frameworks modernes (Next, Nuxt, Vite) le font par defaut. Sur sites legacy ou WordPress, des plugins comme Autoptimize ou WP Rocket peuvent le faire automatiquement.

source · RGESN 4.7 / Google Web.dev - Minify JavaScript #R8-B12

08 ● MINEUR

Des pixels publicitaires sont presents

```
1 pixel(s) marketing : Facebook Pixel.
```

POURQUOI C'EST IMPORTANT

Les pixels marketing (Facebook Pixel, LinkedIn Insight, etc.) sont des trackers dedies a la publicite. Ils ajoutent des kilo-octets de code sur chaque page et pistent les visiteurs.

BONNE PRATIQUE

Ne conserver que les pixels strictement necessaires aux campagnes en cours. Les desactiver entre deux campagnes. A remplacer par du server-side tracking quand possible (Matomo, Plausible server-side).

source · EasyPrivacy #R10-D3

09 ● MINEUR

Votre CSS ne semble pas minifie

```
Votre CSS contient environ 20 % d'espaces et sauts de ligne. Un CSS minifie en contient typiquement moins de 5 %.
```

POURQUOI C'EST IMPORTANT

La minification supprime les espaces, commentaires et sauts de ligne inutiles dans le CSS (et le JS) sans changer son comportement. Sur des fichiers texte, elle economise typiquement 15 a 30 % du poids. C'est une optimisation classique et sans risque.

BONNE PRATIQUE

Activer la minification CSS/JS via votre build tool (Webpack, Vite, Rollup) ou un plugin CMS (Autoptimize, WP Rocket sur WordPress). Passer aussi les fichiers par un CDN qui minifie automatiquement.

source · RGESN 4.7 / Google Web.dev - Minify CSS #R7-B11

10 ● MINEUR

Les images n'ont pas de dimensions explicites

> Ratio : 5% des images ont largeur + hauteur.

POURQUOI C'EST IMPORTANT

Quand les dimensions d'une image ne sont pas connues a l'avance, le navigateur doit faire des calculs de mise en page supplementaires qui consomment du CPU. C'est surtout penalisant sur mobile.

BONNE PRATIQUE

Renseigner les attributs width et height sur chaque balise image. Cela reserve l'espace et evite le decalage de contenu au chargement.

source · Core Web Vitals #R6-B10

11 ● MINEUR

bonne pratique
(sans gain octets)

Aucun CDN detecte pour vos ressources statiques

> 3 sur 48 ressources statiques (CSS, JS, images, fonts) servies depuis un domaine tiers. Un CDN rapprocherait ces ressources de vos visiteurs.

POURQUOI C'EST IMPORTANT

Un CDN (Content Delivery Network) replique vos ressources statiques sur des serveurs repartis geographiquement. Cela ameliore la latence et la resilience pour vos visiteurs eloignes du serveur d'origine. Bonne pratique sans impact direct sur le poids transfere : le modele Sustainable Web Design v4 que nous appliquons utilise une intensite energie constante par octet, independamment du chemin reseau. Le gain CO2 d'un CDN n'est pas mesurable de facon defendable - donc le score d'audit ne bougera pas a la mise en oeuvre. Les benefices reels (cache HTTP, compression Brotli, HTTP/2/3 souvent fournis par les CDN) sont deja captes par les recommandations B1 (compression) et B4 (cache long).

BONNE PRATIQUE

Activer un CDN sur vos ressources statiques (Cloudflare, Bunny, Fastly, CloudFront). La plupart proposent une configuration gratuite ou a faible cout. Cloudflare se met en place en quelques minutes avec un changement de DNS.

source · GR491 UXC-13 #R11-E2

12 ● MINEUR

Des animations tournent en boucle sur votre page

```
3 animation(s) en boucle detectee(s) dans le CSS.
```

POURQUOI C'EST IMPORTANT

Les animations continues maintiennent le processeur graphique actif en permanence - meme quand le visiteur ne regarde pas la zone animee. Sur mobile, cela reduit significativement l'autonomie.

BONNE PRATIQUE

Eviter les animations en boucle sur des elements decoratifs. Les reserver a des micro-interactions ponctuelles. Respecter la preference prefers-reduced-motion pour les utilisateurs sensibles.

source · GR491 UXC-12 #R12-F2

13 ● MINEUR

La preference "moins d'animations" n'est pas respectee

```
Votre CSS ne prevoit pas d'alternative pour les utilisateurs qui ont
active la preference systeme "reduire les animations".
```

POURQUOI C'EST IMPORTANT

Certains utilisateurs (epileptiques, personnes sensibles au mal des transports) desactivent les animations dans les preferences de leur systeme. Respecter ce choix est a la fois une question d'accessibilite et de sobriete energetique.

BONNE PRATIQUE

Ajouter une regle CSS @media (prefers-reduced-motion: reduce) qui desactive ou ralentit les animations quand la preference est active.

source · WCAG 2.3.3 #R13-F4

/METHODOLOGIE

Comment ces chiffres sont calcules

Modele CO2 applique

Les emissions sont calculees selon le modele **Sustainable Web Design v4** (Wholegrain Digital + Green Web Foundation, 2024). Formule : `energie_kWh = bytes / 1e9 * 0.81` puis `co2 = energie_kWh * intensite_carbone`. L'energie est repartie en **22% datacenter / 24% reseau / 54% device utilisateur**.

Poids retenu pour le calcul : on n'utilise pas le poids brut transfere (= phase 2 du tableau /crawl) mais une **moyenne ponderee** qui tient compte du lazy loading et du cache navigateur. Sur ce site : poids retenu = `1.5 Mo` (soit 32 % de moins que le poids transfere 2.2 Mo). Detail dans la page methodologie publique.

Intensites appliquees dans ce rapport (audience `France`) :
• datacenter : `50 gCO2/kWh` (hebergeur vert, Green Web Foundation)
• reseau + device : `57 gCO2/kWh`

Ratio cachable calcule automatiquement depuis les entetes Cache-Control des ressources statiques : `0%`. Ratio visiteurs retour : `25%`.

Fourchette basse / haute

Le coefficient `0.81 kWh/Go` est conteste dans la litterature. Aslan et al. (2018) et Coroama et al. (2021) proposent `0.06 a 0.20 kWh/Go`. Nous affichons donc systematiquement une **fourchette basse/haute** dans le bloc CO2 pour refleter cette incertitude methodologique.

Acquisition des donnees

Un navigateur Chromium headless (Playwright) charge la page comme un visiteur reel, execute le JavaScript, attend la stabilisation du reseau, et capture chaque ressource telechargee. Chaque fichier est type, pese, classe (first-party ou tiers), et son format et ses entetes HTTP sont analyses. Les analyseurs exploitent ensuite ces donnees pour generer les 47 criteres.

Lazy loading : double mesure initial vs max

Beaucoup de sites modernes chargent les images et les iframes "a la demande" quand elles entrent dans le viewport (`loading="lazy"` ou IntersectionObserver). Mesurer uniquement la page a l'arrivee sous-estime le poids reel pour un visiteur qui parcourt la totalite. Notre crawler procede donc en **deux temps** :

Phase 1 - chargement initial : la page est ouverte, le JS s'execute, on attend la stabilisation du reseau (networkidle). On capture les octets et requetes a ce moment. Cette mesure correspond a un visiteur qui reste en haut de page.

Phase 2 - auto-scroll : on deroule la page de haut en bas par paliers pour declencher les IntersectionObserver. On attend a nouveau la stabilisation reseau. Cette mesure (worst-case) represente un visiteur qui fait defiler toute la page.

Le **calcul CO2 utilise une phase 3 (poids retenu)** : moyenne ponderee 50/50 phase 1 / phase 2, puis ajustement pour le cache navigateur des visiteurs qui reviennent. Resultat : le lazy loading bien fait *recompense* le score (le poids retenu baisse), et le worst-case (phase 2) reste affiche dans la section /crawl pour la transparence et pour la comparaison avec HTTP Archive ou Website Carbon. Detail dans la page methodologie publique.

Reactivite serveur (TTFB)

Le **TTFB** (time to first byte) mesure le temps ecoule entre la requete HTTP et le premier octet recu du serveur. C'est un indicateur observable qui combine la reactivite serveur et la latence reseau.

Il est affiche dans la section "hebergement" a titre **informatif seulement**, **non integre au score CO2**, et **non interprete automatiquement** par nos soins. Reperes usuels cites dans les publications sur les Web Vitals : < 200 ms (bon), 200-600 ms (acceptable), > 600 ms (lent). L'interpretation d'un TTFB eleve (infrastructure, cache, application, reseau) necessite une analyse contextuelle que cet audit front ne permet pas.

Limite methodologique : charge serveur applicative

Le modele **Sustainable Web Design v4** que nous appliquons rapporte la charge datacenter (22% du total) au **poids transfere**. Cette approche ne differencie pas un serveur qui sert du HTML statique d'une application web lourde (CMS, e-commerce avec beaucoup de requetes DB) qui executerait beaucoup de code pour produire une page de meme poids.

Concretement : a poids de page egal, deux sites peuvent avoir une consommation serveur reelle tres differente selon leur stack applicative. Notre modele ne capture pas cette difference.

Mesurer rigoureusement la charge applicative necessiterait un acces au serveur (instrumentation CPU, RAM, requetes DB), impossible depuis un audit front. Il existe tres peu d'etudes scientifiques comparables et reproductibles sur la consommation serveur des CMS / frameworks. Plutot que de proposer une classification non etayee, nous **assumons cette limite** et ne tirons aucun verdict automatise sur la stack utilisee. Le TTFB reste le seul signal observable cote client.

Sources et referentiels

Chaque critere est ancre sur un referentiel public :

- **RGESN v1** (ARCEP / ADEME / DINUM, 2024)
- **GR491** (Institut du Numerique Responsable)
- **W3C Web Sustainability Guidelines**
- **Sustainable Web Design v4** (modele CO2)
- **Base Carbone ADEME** (facteurs France 2024)
- **Green Web Foundation API** (verification hebergeur vert)
- **WhoTracks.me + EasyPrivacy** (liste trackers)

Chaque recommandation cite la source precise qui la fonde.

Limites assumees

Nous mesurons le comportement au premier chargement, **sans consentement cookies**. Certains sites bloquent les trackers avant acceptation du CMP, ce qui peut sous-estimer leur empreinte reelle. Les sites qui chargent plus de contenu apres interaction utilisateur ne sont pas completement analyses. Le modele device est une approximation statistique (54% de la consommation totale).

/PROCHAINE ETAPE

Pas envie d'appliquer tout seul ?
On s'en occupe.

Carboskale est edite par **Nature Digitale**, studio d'eco-conception base en France. On peut vous accompagner sur les chantiers identifies ci-dessus, former votre equipe ou auditer plus en profondeur.

contact@naturedigitale.fr · 02 19 00 16 45 · prendre rendez-vous

NOUS ECRIRE →

© 2026 nature digitale · carboskale v0.1 · fr