





FAUT-IL OPTIMISER LES VÉHICULES ?

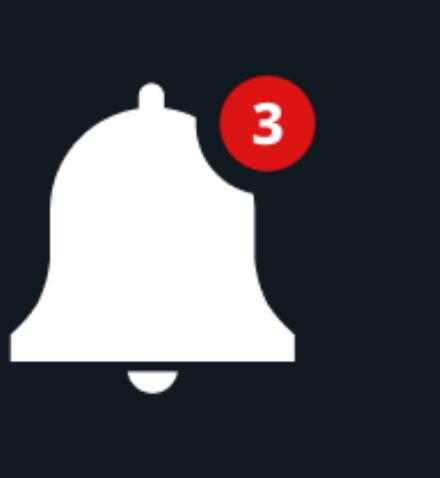


OU MODIFIER LA CONSOMMATION ?



FAUT-IL OPTIMISER LES MACHINES ?

```
<li><a href="index.html">Home</a></li>
<li><a href="home-events.html">Home Events</a></li>
<li class="has-children"> <a href="#" class="current">Header Options</a>
  <ul>
    <li><a href="tall-button-header.html">Tall Button Headers</a></li>
    <li><a href="image-logo.html">Image Logo</a></li>
    <li class="active"><a href="tall-logo.html">Tall Logo Images</a></li>
  </ul>
<li class="has-children"> <a href="#">Carousels</a>
  <ul>
    <li><a href="variable-width-slider.html">Variable Image W...
    <li><a href="testimonial-slider.html">Testimonial Slider<...
    <li><a href="featured-work-slider.html">Featured Work SL...
    <li><a href="equal-column-slider.html">Equal Column Slid...
    <li><a href="video-slider.html">Video Slider</a></li>
    <li><a href="mini-bootstrap-carousel.html">Mini Slider</li>
  </ul>
```



OU MODIFIER LOGICIELS ET UTILISATION ?

LE CÔTÉ OBSCURE



Production de gaz à effet de serre



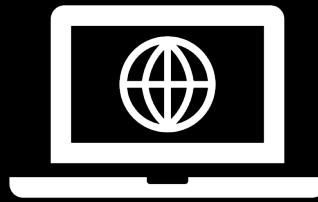
2019
4%

Le numérique consomme
de plus en plus

2025
8%

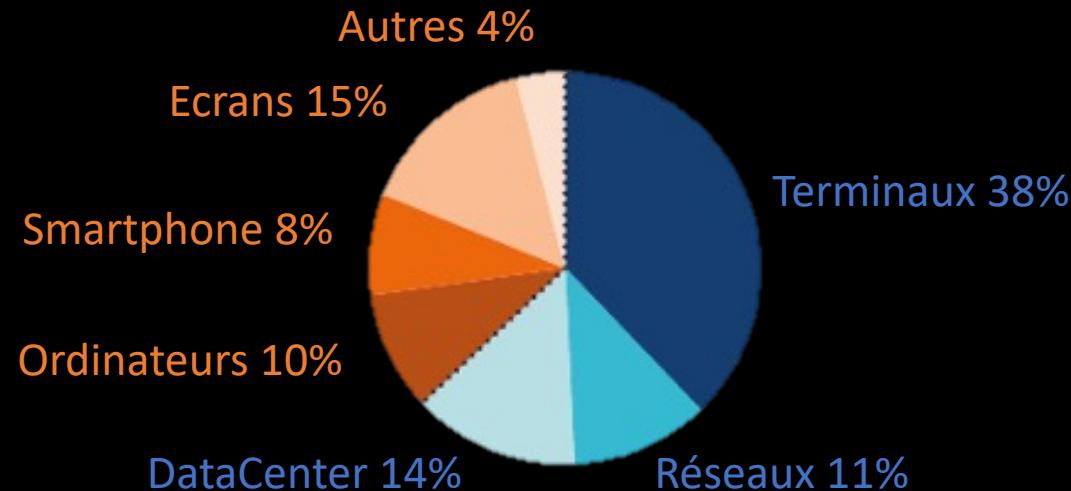


9%



Un laptop de 2 kg c'est

- 103kg de CO₂
- 1,5 tonnes d'eau
- 600kg de matière première
- 70 matériaux



Production (37%)

Empreinte carbone du numérique (2019)

Utilisation (63%)



Ma consommation web

- 7g de CO₂/recherche
- 4g CO₂/mail
- 500 milliards mail/j
- 135kg CO₂ mail/employés



Streaming = 60% du trafic
Dont 27% pour le XXX

3 AXES À ADRESSER



Hardware

L'ingénieur

Politique d'achat durable
Réduire la consommation
Changer le mix énergétique
Réparer au lieu de jeter
Réutiliser ou recycler

Software

Le codeur

Réutiliser avant de coder
Analyser impact énergétique
Concevoir selon bonnes pratiques
Choisir les bonnes technologies
Optimiser les usages

Sobriété

L'utilisateur

Sensibiliser les utilisateurs
Réduire les achats/installations
Réduire l'utilisation
Paramétrier pour optimiser

AXE SOFTWARE (DEVELOPPEMENT DURABLE)

- Dans le software, c'est surtout l'exécution qui nous intéresse, les impacts écologiques de production ou d'élimination sont en général ridicule à côté des impacts d'exploitation.
- Ainsi, il faut utiliser les bonnes pratiques de design, choisir les bonnes technologies, etc.
 - A. Les choix d'utilisation
 - B. Les choix de design
 - C. Les choix techniques
 - D. Les choix d'exploitation

CHOIX D'UTILISATION : LESS IS MORE

- Réduire les fonctionnalités à l'essentiel
- Réduire le nombre d'applications (reuse before code)
- Simplifier les interfaces utilisateurs
- Pas d'alertes par défaut, paramétrage pour ajouter
- Accepter d'attendre plutôt que tout charger au cas où
- Version simplifiée pour impression
- Des images statiques plutôt qu'animée ou que des vidéos
- Ne pas faire «play» sans ordre utilisateur
- Limiter l'utilisation des capteurs et ressources des smartphone
- Limiter l'intégration de services tiers

CHOIX DE DESIGN

- Choix d'architecture modulaire et scalable
- Reuse before build
- Réduire les flux de données
 - alléger les médias, gestion du cache, compression
 - local vs connecté, éviter les boucles, requête par pages
- Réduire le volume des données stockées
 - minimalisme
 - archivage et épuration
- Réduire les schedules et notifications
- Réduire les effets de l'obsolescence HW ou SW

C

CHOIX TECHNIQUES

- Choix des langages
- Choix des frameworks/library
- Choix des middlewares (serveur web asychrone et multithread)
- Architecture asynchrone vs synchrone
- Bonnes pratiques de codage (réduire la taille, mutualiser fonctions, etc.)

QUELS LANGAGES UTILISER ?

- La consommation est très différente selon les types de langages :
 - compilés 120 J
 - C, C++, Rust, Go
 - machine virtuelle 576 J
 - Java, C#
 - interprétés 2365 J
 - PHP, Ruby, Python, JS

[Energy Efficiency Across Programming Languages \(2017\)](#)

	Energy	Time		Mb	
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) JRuby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) JRuby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) JRuby	19.84

CHATGPT : C'EST QUOI LES BONNES PRATIQUES ?

1. Optimisation de l'efficacité algorithmique
2. Éviter le Polling
3. Minimisation des requêtes réseau
4. Utilisation de la mise en cache
5. Code mort et réduction de la taille du code
6. Utilisation efficace des threads et de la concurrence
7. Optimisation des ressources graphiques
8. Adaptabilité du matériel
9. Gestion de l'état de l'application
10. Tests et monitoring

CHOIX D'EXPLOITATION BY DESIGN

- Niveau de log
- Nbr environnements adaptés à la charge
- Environnements test/dev pas allumés 24/7

DURABLE : PAS JUSTE ÉCOLOGIQUE

- Intégrer les besoins spécifiques des personnes en situation de handicap (normes d'accessibilité)
- Respect des règles d'éthique
- Opensource ?
- Protection des données
- Documenter pour réutiliser

QUE FAIRE EN TANT QUE LIEU DE FORMATION ?

Savoir-faire

- Sensibiliser les étudiant.e.s (conférences)
- Présenter les bonnes pratiques (workshops)
- Créer du plaisir à optimiser les codes (hackathon)

Faire-savoir

- Démontrer l'impact économique (blueprint)
- Souligner les effets de bord (impact factor)
- Communiquer aux entreprises (social network)

CONCRÈTEMENT ?

Nous souhaitons réaliser un «Green Hackathon»

- Des entreprises proposent des codes existants avec des métriques (nombre d'utilisation, volumétrie, etc.)
- Des étudiant.e.s peuvent utiliser leurs propres projets
- On mesure l'impact énergétique du code
- On applique certaines bonnes pratiques et on remesure à chaque fois
- On finalise avec des propositions chiffrées en termes d'impact énergétique => électrique => CHF

