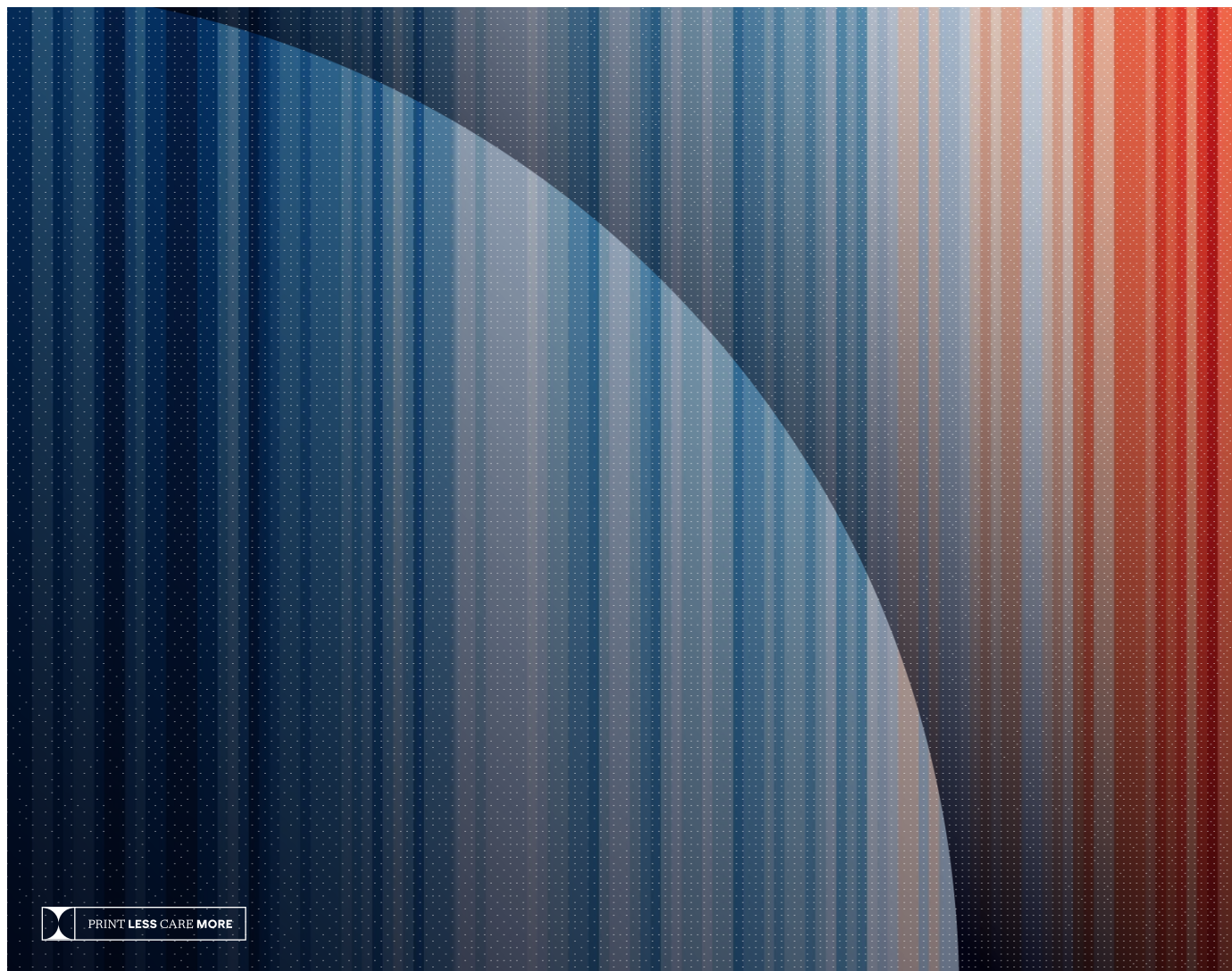
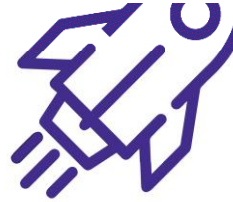


Référentiel d'Ecoconception Davidson

150 pratiques pour minimiser l'impact
environnemental des services numériques





MANIFESTE POUR L'ÉCOCONCEPTION DES SERVICES NUMÉRIQUES

Les services numériques sont un levier extraordinaire pour transformer nos sociétés mais leur développement, contribuant au dépassement des limites planétaires connues, accroît l'exploitation et le risque d'extinction du vivant (espèce humaine incluse) lié à l'ère de l'Anthropocène.

Ce manifeste engage ses signataires à repenser leur manière de concevoir ces services et mettre en œuvre tous les moyens permettant d'y parvenir en s'appuyant sur 4 grands principes pour guider cette démarche :

- *Des services numériques répondant aux justes besoins de nos utilisateurs, plutôt que d'en créer de nouveaux.*
- *Une vision en cycle de vie complète et systémique, plutôt qu'une approche fragmentée en assemblage d'éléments isolés.*
- *Des plateformes ouvertes favorisant la réutilisation des données et des algorithmes, plutôt que la capitalisation sur des systèmes fermés.*
- *La transparence et la communication sur les impacts environnementaux, plutôt que l'ignorance ou la dissimulation de ces impacts.*

Nous acceptons que :

- Tout nouveau système technologique aura des impacts négatifs sur l'environnement, même si celui-ci a été développé en suivant les meilleures pratiques d'écoconception.
- Chaque choix de conception aura une conséquence sur l'empreinte environnementale de notre service, et nous devons évaluer et suivre ces conséquences dès le début du processus de création.
- Les efforts pour diminuer l'empreinte environnementale de nos services sont continus et évolutifs, et nous nous engageons à progresser à chaque étape de la conception.
- Le potentiel de réduction des impacts liés à d'autres secteurs d'activité par la technologie ne va pas de soi et doit être démontré par une modélisation rigoureuse des impacts évités.
- Il est de notre entière responsabilité en tant que producteurs de service numérique de ne pas créer de sentiment d'obsolescence pouvant mener à un renouvellement prématuré des équipements de nos utilisateurs.
- L'écoconception nécessite un engagement collectif, intégrant les équipes, les parties prenantes ET les utilisateurs.

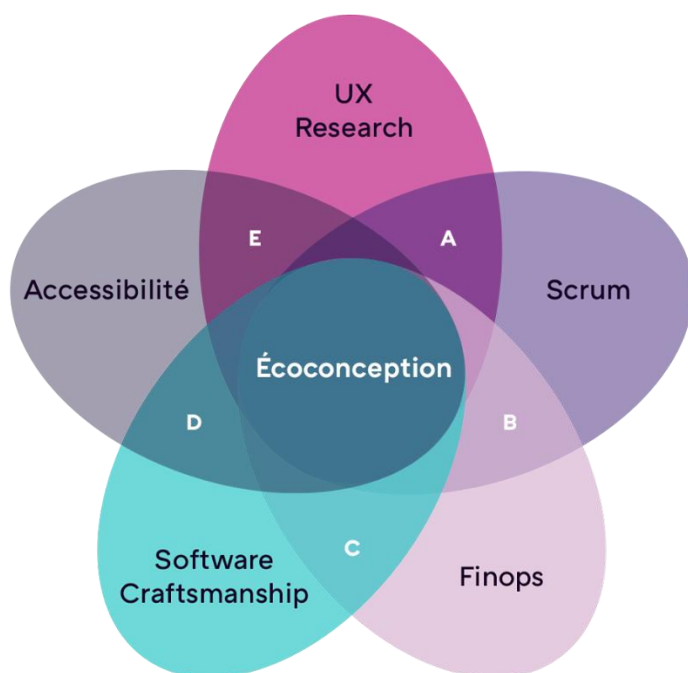
Nous appelons à la responsabilité des acteurs du numérique et à une transformation radicale des pratiques pour en contenir les impacts négatifs. Ainsi ce manifeste affirme notre engagement à intégrer l'écoconception dans la création, l'utilisation et la fin de vie des services numériques.



LES CO-BENEFICES DE L'ÉCOCONCEPTION

Au-delà des considérations environnementales qui justifient à elles seules des changements drastiques de certaines pratiques de conception et la mise en avance de la sobriété numérique (cf [scenarios de l'ADEME ARCEP pour 2030 et 2050](#)), l'écoconception a l'immense avantage de s'aligner avec des paradigmes de développement déjà existants (et pour certains biens installés), qui font rapidement apparaître des co-bénéfices qui la rende vraiment intéressante sur plusieurs aspects.

Voici notre vision de ses imbrications :



- A** | Délivrer le service remplissant précisément le besoin de l'utilisateur, ni plus, ni moins.
- B** | Livraison continue des fonctionnalités en production sur une infrastructure parfaitement dimensionnée.
- C** | Code source parfaitement optimisé et maintenable nécessitant une infrastructure minimaliste et la moins chère possible.
- D** | Design for all, utiliser son talent d'artisan du code pour rendre le service accessible à tous.
- E** | Expérience utilisateur inclusive et parfaitement adaptée à tous.

Nous avons donc une excellente nouvelle : **Si vous appliquez déjà certaines de ces pratiques dans vos projets, vous possédez déjà certains réflexes permettant de penser l'écoconception.** Par conséquent, certaines pratiques exposées dans ce référentiel vous sembleront parfaitement connues et déjà en place.

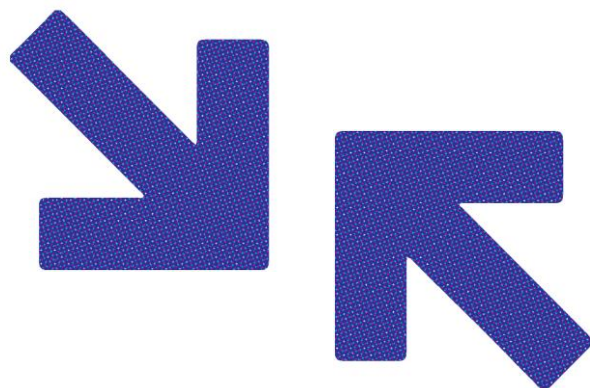
Félicitations, la moitié du chemin est déjà fait !

Cependant, l'application isolée des démarches Scrum, FinOps ou d'UX ne garantissent pas forcément qu'un service numérique aura minimisé son impact environnemental. Par exemple, un service cloud hébergé dans une région au mix électrique très carboné pourra par moment être moins cher que s'il est situé dans une région au mix électrique faiblement carboné, nous aurons économisé un peu sur la facture (FinOps), mais plus fortement contribué aux changements climatiques.

C'est bien l'application de tous ces paradigmes, alignés avec une vision systémique des impacts de nos choix de conceptions, qui pourra faire en sorte que notre produit numérique puisse s'intégrer dans une trajectoire de soutenabilité globale.

Adopter l'écoconception, c'est s'assurer de créer un produit répondant aux justes besoins de ses utilisateurs (Scrum), sur des parcours sobres, efficaces (UX), inclusifs et minimisant la facture numérique (Accessibilité), minimisant les coûts d'hébergement (FinOps), de développement (Scrum), et de maintenance (Software Craftmanship), et qui en plus de tout ça limite au maximum ses externalités négatives sur l'environnement. Et oui, tout ça à la fois !

SI VOUS AVEZ LA FLEMME DE TOUT LIRE



ON LE FAIT EN UNE PAGE

Ce dont on ne parle pas (la démarche de Numérique Responsable), mais qu'il faut faire quand même :

1. Nommer un responsable numérique frugal
2. Réaliser une analyse de cycle de vie complète de la direction des services technologiques
3. Réduire le taux d'équipement (gare aux seconds écrans en particulier !)
4. Acheter reconditionné
5. Allonger la durée de vie de tous les équipements, réparer quand c'est possible
6. Donner pour réutilisation, reconditionnement, et recyclage en dernière option

Ce que l'on détaille dans nos pratiques :



Limiter le gras
fonctionnel



Diminuer les volumes de
stockage



Penser inclusivité



Limiter les échanges de
données, les requêtes



Mettre en cache tout ce
qu'on peut



Vérifier PUE, WUE et CUE
des datacenters



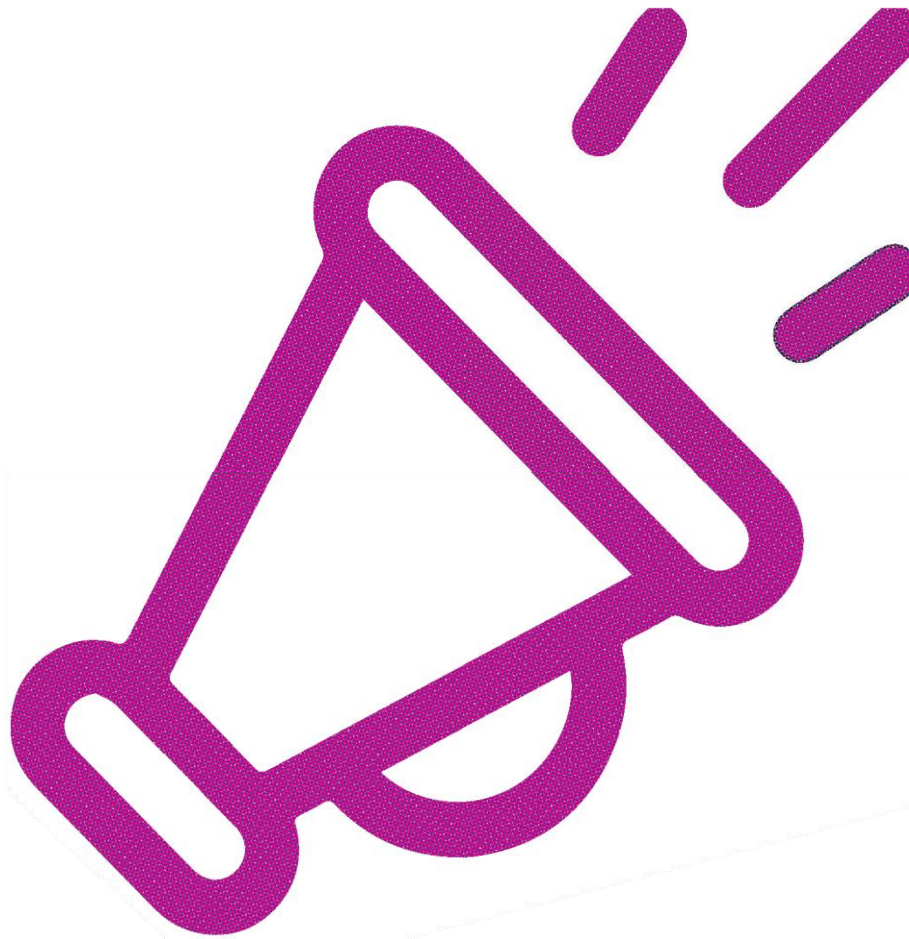
Limiter l'usage de l'IA



Penser sur tout le cycle de
vie de nos services



Concevoir des interface
sobres et compatibles



AVANT TOUTE CHOSE

SENSIBILISATION

Avant de nous plonger dans les bonnes pratiques, arrêtons-nous quelques instants sur ce qui nous semble être la clé pour bien démarrer et mettre l'écoconception au cœur de nos projets numériques : la sensibilisation des parties prenantes (collaborateurs, clients, partenaires) à la question écologique.

Si les femmes et les hommes qui contribuent au projet n'ont pas compris l'importance de la démarche, sa mise en œuvre n'en sera que plus longue !

Dans un futur proche, Internet pourrait devenir la première source mondiale de pollution.

Sous son apparente immatérialité, le numérique est au contraire on ne peut plus **matériel**, et ses impacts sur l'environnement également. En France, le numérique représente déjà **4,4% de l'empreinte carbone en 2022, soit 29,5 MtCO₂e** (chiffres de la mise à jour 2025 de l'étude [ADEME - Numérique & environnement : entre opportunités et nécessaire sobriété](#)). Si la tendance continue, on s'attend à un **triplément des émissions de Gaz à Effet de Serre (GES) du numérique d'ici 2050**.

Si on considère les 3 tiers d'un service numérique, **les terminaux, le réseaux internet et les centres de données**, on constate que **50% des émissions de GES du numérique** sont liés aux **terminaux** des utilisateurs (PC, smartphone...) et **46%** sont liés aux **centres de données**. Dans un pays possédant un [mixin électrique à faible émission de GES](#), 80% de l'impact d'un équipement sera lié à **la phase de fabrication** (sauf pour les serveurs qui tournent en permanence et sont plutôt sur un ratio de 50/50).

A cela s'ajoute les impacts sur la consommation d'électricité, l'utilisation d'eau douce, la déplétion de métaux, les pollutions chimiques, l'utilisation des sols... sans oublier tous les impacts en aval que les services numériques généreront par leur utilisation dans d'autres secteurs d'activité.

Sachant que 80% des impacts d'un service sont déterminés pendant sa phase de conception, pour Davidson les quatre leviers clé de l'écoconception seront alors de :

1. limiter au maximum la quantité de ressources matérielles nécessaires pour faire fonctionner nos services ;
2. ne pas rendre obsolète les équipements des utilisateurs de nos services ;
3. porter une attention particulière à l'hébergement de nos services et l'utilisation de services tiers ;
4. produire des services œuvrant pour le bien commun ;



Pour bien démarrer, nous allons avoir sélectionné pour vous quelques ressources utiles !

DÉBUTANT

Lectures & mooc

Fresque du climat

<https://fresqueduclimat.org/>

Fresque du numérique

<https://www.fresquedunumerique.org/>

Les scénarios de l'ADEME-Arcep sur l'empreinte environnementale du numérique en 2020, 2030 et 2050

<https://www.arcep.fr/la-regulation/grands-dossiers-thematiques-transverses/lempreinte-environnementale-du-numerique/etude-ademe-arcep-empreinte-environnemental-numerique-2020-2030-2050.html>

CONFIRMÉ

Les rapports du Shift Project sur les impacts environnementaux du numérique

<https://theshiftproject.org/lean-ict/>

Le rapport de France Stratégie «La consommation de métaux du numérique»

<https://www.strategie.gouv.fr/sites/strategie.gouv.fr/files/atoms/files/fs-2020-dt-consommation-metiaux-du-numerique-juin.pdf>

Vidéos, podcast & mooc

Podcast du numérique responsable MINumEchos

<https://ecoresponsable.numerique.gouv.fr/publications/podcast-minumechos/>

🕒 30 minutes

MOOC de sensibilisation de l'INR

<https://www.academie-nr.org/sensibilisation/#/>

🕒 30 minutes

Impacts environnementaux du numérique de l'INRIA

<https://youtu.be/x3vf4mxVXNA?si=nWVaaV-QEPiUTvfv>

🕒 30 minutes

Le MOOC impacts environnements de l'INRIA

<https://www.fun-mooc.fr/fr/cours/impacts-environnementaux-du-numerique/>

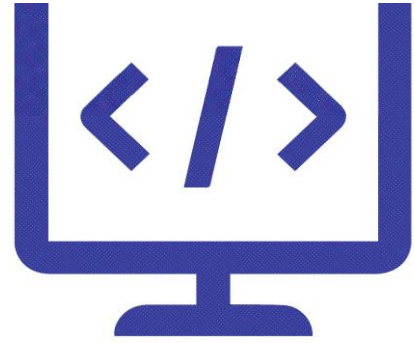
<https://learninglab.gitlabpages.inria.fr/mooc-impacts-num/mooc-impacts-num-ressources/>

🕒 4 heures

Le MOOC complet de l'INR

<https://www.academie-nr.org/#mooc-nr>

🕒 4 heures

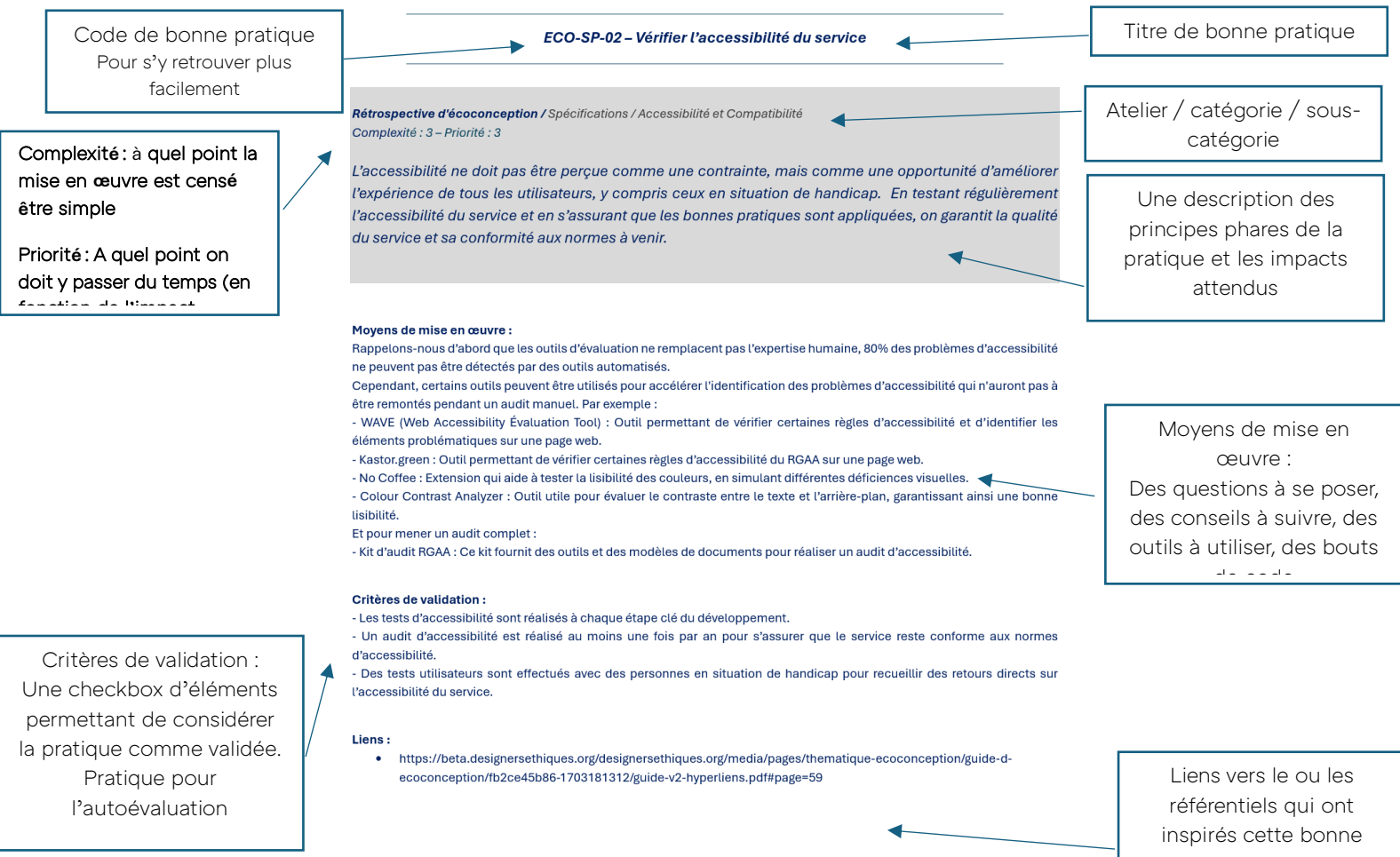


LES BONNES PRATIQUES

Les bonnes pratiques que vous allez trouver dans ce référentiel sont issues d'un travail de centralisation et d'adaptation mené sur les 5 référentiels suivants que nous considérons comme les plus pertinent dans leur domaine.

Référentiel	Public	Lien
RGESN (2024)	Product Manager / Product Designer / Product Owner / Architecte / UX / Ops	Lien
Guide des designers éthiques	Product Manager / Product Designer / Product Owner / UX / Éditeur de contenus	Lien
API Green Score	Architecte / Développeur Backend	Lien
115 bonnes pratiques d'écoconception web	Product Owner / Développeur Frontend / Éditeur de contenus	Lien
Référentiel Général pour l'IA Frugale	Architecte / Ops / Data Scientist	Lien

Chaque bonne pratique est matérialisée par une fiche composée des éléments suivants :



Afin de faciliter son exploration, le référentiel d'écoconception Davidson est découpé en sections représentant les moments où les choix de conception associés ont le plus de probabilité d'apparaître (par atelier ou par cérémonies).

Cérémonies / Atelier	Définitions
AB: Ateliers de Brainstorming	Ateliers au cours desquelles l'équipe Scrum génère un grand nombre d'idées afin d'explorer différentes solutions à un problème ou à une opportunité, en vue d'enrichir le backlog produit.
UX: Ateliers UX/UI	Ateliers au cours desquelles l'équipe Scrum, en collaboration avec les designers UX/UI, explore, conçoit et affine l'expérience utilisateur d'une fonctionnalité ou d'un produit, en s'appuyant sur les spécifications du produit et les retours utilisateurs.
AR: Ateliers d'architecture logiciel, devops, hébergement	Ateliers au cours de laquelle l'équipe Scrum, en collaboration avec les architectes, explore et définit les grandes lignes techniques du produit, en s'assurant de leur alignement avec les besoins fonctionnels et les contraintes du projet.
SP: Sprint Planning	Cérémonie au cours de laquelle l'équipe Scrum définit un objectif clair pour un sprint à débiter, sélectionne les US à réaliser et planifie la manière dont elles seront réalisées.
SR: Sprint Review	Cérémonie au cours de laquelle l'équipe Scrum présente l'incrément de produit réalisé durant le sprint aux parties prenantes, recueille leurs retours et ajuste le backlog produit en conséquence.
ECO: Rétrospective d'écoconception	Cérémonie à mettre en place, régulière (6 mois / 1 an), spécifiquement dédiée à l'analyse des impacts des services et à l'étude des possibilités d'améliorer l'écoconception.

Chaque bonne pratique est catégorisée par champs d'expertise. Ainsi, au sein de chaque section, vous n'avez plus qu'à vous concentrer sur celles que vous êtes le plus à même de prendre en charge dans votre position d'expert et au quotidien.

Catégories	Définitions
ST : Stratégie	La stratégie permet de déterminer et de suivre la pertinence, les enjeux et le pilotage de la conception du service numérique
SP : Spécifications	Cette partie regroupe les éléments du cadrage du projet sur toute la durée du service numérique
AR : Architecture	Cette partie porte sur la stratégie de conception et l'articulation des composants applicatifs entre le frontend et le backend
UX : UX/UI	Cette partie présente les étapes et méthodes de conception des services numériques pour définir les meilleures solutions d'interaction destinées aux utilisateurs.
CO : Contenus	Cette partie concerne les documents et médias informatifs ajoutés au service numérique par des personnes contributrices et disponibles pour l'utilisateur final
FE : Frontend	Le frontend est l'ensemble des composants en opération sur un terminal utilisateur pour permettre l'utilisation d'un service numérique
BE : Backend	La partie s'intéresse à l'ensemble des composants en opération coté serveur pour permettre le fonctionnement d'un service numérique
HE : Hébergement	Il s'agit des moyens mis en œuvre coté hébergement pour permettre la conception / développement, l'utilisation, et si applicable, l'entraînement d'un service numérique. Toute la chaîne d'hébergement mobilisée pour les fonctionnalités critiques du service (centres de données, Content Delivery Network, etc.) doit être prise en compte pour valider les critères de cette partie.
IA : Intelligence Artificielle	Cette partie concerne les services numériques reposant sur une intelligence artificielle (IA). Elle vise la mise en place de principes d'écoconception et de frugalité quant à l'entraînement et l'inférence des modèles algorithmiques utilisés pour l'IA. La phase d'apprentissage désigne le processus par lequel un système réalise, à partir de données et via des modèles algorithmiques, des calculs afin de proposer des fonctionnalités. Elle est suivie par une phase d'inférence, de mise en œuvre des modèles entraînés.
API : API	Cette partie concerne l'interface d'échange de données entre le backend et le monde extérieur, qui peut être constitué de frontend, d'applications mobiles ou de services externes consommateurs du service.

Nul besoin de connaître ce référentiel sur le bout des doigts, il s'agit plutôt d'un outil d'aide à la conception que nous vous encourageons à garder disponible, et à utiliser dès que l'occasion se présente. Quand vous participez à un atelier de conception, ouvrez ce référentiel à la section appropriée et rendez-vous à la sous-section relative à votre rôle dans la conception, et demandez-vous régulièrement si les choix que vous prenez sont conformes à l'esprit de la démarche d'écoconception décrite dans ce guide.

Rappelez-vous que tout est toujours une question d'arbitrage et qu'aucun système ne peut être parfait, mais que cela ne fait pas de mal de viser le beau résultat.

SOMMAIRES DES BP

MANIFESTE POUR L'ECOCONCEPTION DES SERVICES NUMÉRIQUES	1
LES CO-BENEFICES DE L'ECOCONCEPTION	2
SI VOUS AVEZ LA FLEMME DE TOUT LIRE	3
AVANT TOUTE CHOSE	4
Lectures & mooc Vidéos, podcast & mooc.....	6
LES BONNES PRATIQUES.....	7
SOMMAIRES DES BP	10
Ateliers de Brainstorming.....	11
Rétrospective d'écoconception.....	17
Ateliers UX/UI.....	27
Ateliers d'architecture logiciel, devops, hébergement.....	63
Sprint Planning	112
Sprint Review	165
ANNEXES.....	168

Ateliers de Brainstorming

Ateliers au cours desquelles l'équipe Scrum génère un grand nombre d'idées afin d'explorer différentes solutions à un problème ou à une opportunité, en vue d'enrichir le backlog produit.

*AB-SP-01 – Évaluer le service en termes d'utilité dans la bifurcation
environnementale ou les autres objectifs de développement durable (ODD)*

Ateliers de Brainstorming / Spécifications / Utilité du service

Complexité : 3 – Priorité : 3

Prendre en compte l'utilité du service numérique dès sa conception, par exemple son inscription dans au moins l'un des objectifs de développement durable (ODD), l'un des enjeux de limites planétaires ou tout autre référentiel du même type.

Il s'agit de déterminer en amont du projet si l'utilité du service est avérée parmi :

- *Les 17 objectifs de développement durable (ODD) de l'ONU ;*
- *Les 9 limites planétaires ;*
- *La directive CSRD – Corporate Sustainability Reporting Directive ;*

Si l'utilité du service ne s'inscrit pas dans ces référentiels, justifier en quoi le service est utile, participe à l'intérêt général ou est en appui d'une politique publique.

Moyens de mise en œuvre :

Le rapport comprendra a minima une réponse aux questions suivantes (source : RGEN et Designers Éthiques) :

- L'utilisation du numérique pour ce service est-elle nécessaire ?
- Existe-t-il d'autres solutions non numériques pour répondre à ce besoin ?
- Quels sont les réels besoins justifiant la création du service ?
- La valeur ajoutée du service justifie-t-elle la mobilisation des ressources requises pour sa création ? Est-ce qu'on crée plus de valeur qu'on en détruit ?
- Pour chaque fonctionnalité, est-elle vraiment nécessaire ? Peut-on faire autrement ?
- Que se passerait-il si on ne l'avait pas ?
- Le service répond-t-il à des besoins essentiels ?
- Le service participe-t-il à la mise en place de communs numériques ?

Il convient d'afficher dans la déclaration d'écoconception les ODD dans lequel le service s'inscrit

Critères de validation :

- La déclaration d'écoconception du service numérique explique comment a été évalué le service, par exemple dans quels objectifs de développement durable il s'inscrit, quelles sont les réponses apportées aux enjeux de limites planétaires ou autre référentiel utilisé (préciser lequel), et leur pertinence.
- Le service a correctement évalué via une analyse quantitative des impacts environnementaux et sociaux directs et indirects potentiels qu'ils génèrent, et justifie ces impacts par les bénéfices qu'il apporte. Voir la méthode Empreinte Projet niveau 1 de l'ADEME.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=20

Ateliers de Brainstorming / Spécifications / Définition du besoin

Complexité : 3 – Priorité : 3

Avant de se lancer dans la conception d'un service numérique, il est crucial de bien cerner les besoins réels pour éviter de concevoir des fonctionnalités superflues qui alourdiront à la fois le service et son empreinte environnementale à chaque utilisation. Environ 45% des fonctionnalités demandées ne sont jamais utilisées, et 70% ne sont pas essentielles (source : Les 115 bonnes pratiques • GreenIT). La définition du besoin demande un regard critique quant à la réelle utilité des fonctionnalités et doit se faire avec toutes les parties prenantes du projet au plus tôt dans le processus de conception. En se concentrant sur l'essentiel, on réduit les besoins en ressources informatiques et on simplifie l'utilisation du service.

Moyens de mise en œuvre :

- Posons-nous les bonnes questions dès le départ : Quels sont les besoins réels justifiant la création du service ? La valeur ajoutée du service justifie-t-elle les ressources nécessaires à sa création ? Cette fonctionnalité est-elle vraiment nécessaire ? Que se passerait-il si on ne l'avait pas ?
- Détectons les fonctionnalités non essentielles au moment de l'analyse de l'expression du besoin. La méthode MoSCoW, des ateliers, des wireframes ou des prototypes avec tests utilisateurs permettent de vérifier l'utilité d'une fonctionnalité en amont de son développement.
- Utilisons les principes du Green UX pour rester centré sur les besoins réels et les attentes des utilisateurs. Aller à l'essentiel du besoin ne va pas au détriment de l'expérience utilisateur, bien au contraire. Appliquer une approche "Green-UX" a des conséquences sur les décisions concernant la composition d'une interface, le choix de composants et de fonctionnalités, le parcours de navigation. Cela se répercute automatiquement sur les métiers du design, du développement, d'architecture et de la maintenance du service.
- Utilisons le Impact Agile Poker pour estimer facilement l'impact d'une fonctionnalité sur le service <https://www.suricats-consulting.com/impact-poker-agile-engage/>
- Inspirons-nous des succès du Web qui misent sur une grande sobriété fonctionnelle. Respectez le principe YAGNI (You Ain't Gonna Need It) de l'eXtreme Programming : développez une fonctionnalité seulement quand elle est vraiment nécessaire.

Critères de validation :

- L'expérience utilisateur (UX) est centrée sur les besoins réels et les attentes des utilisateurs.
- Les fonctionnalités retenues ont toutes été analysés au regard de leurs impacts sur le service.
- Les décisions concernant la composition de l'interface, le choix des composants et des fonctionnalités, ainsi que le parcours de navigation, sont pris en fonction de principes de sobriété et d'efficacité.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=8>
- Les 115 bonnes pratiques d'écoconception web : 1
- Les 115 bonnes pratiques d'écoconception web : 2
- <https://beta.designersethiques.org/fr/index-conception-responsable>

AB-SP-03 – Définir la liste des profils de matériels que vos utilisateurs vont pouvoir employer pour accéder au service

Ateliers de Brainstorming / Spécifications / Compatibilité matérielle

Complexité : 2 – Priorité : 3

Un service numérique qui exige des ressources techniques dernier cri peut forcer les utilisateurs à renouveler leurs appareils, ce qui est désastreux d'un point de vue de l'impact global du service. Pour éviter cela, il est essentiel de dresser une liste des types de matériel que les utilisateurs devraient pouvoir utiliser, qu'il s'agisse de vieux ordinateurs de bureau, de tablettes, de smartphones ou d'autres dispositifs. Cela inclut des critères comme le débit minimum de la connexion Internet, la taille de l'écran, la puissance du processeur, la mémoire vive, et d'autres spécifications techniques. L'objectif est de garantir que le service soit accessible à un maximum d'utilisateurs, indépendamment de l'âge de leur matériel ou leur type de connexion, tout en anticipant les besoins futurs.

Moyens de mise en œuvre :

Commencer par définir clairement le profil des matériels qui seront supportés par le service. Cela implique de s'assurer que le service soit compatible avec des équipements aussi anciens que possible, afin de minimiser le risque d'obsolescence matérielle. Il peut être utile de réaliser une étude des équipements les plus couramment utilisés par les utilisateurs cibles, afin de mieux comprendre leurs besoins. Ensuite, il faudra établir des spécifications techniques qui pourront être facilement consultées par les utilisateurs.

Critères de validation :

- Le profil des matériels supportés par le service est clairement établi et privilégie le support des équipements les plus anciens possibles.
- Les spécifications minimales requises pour le matériel sont affichées dans la déclaration d'écoconception du service, de manière à être facilement accessibles aux utilisateurs.
- Si certaines fonctionnalités du service nécessitent des versions de matériel plus récentes, ces versions minimales doivent être clairement indiquées, avec une distinction entre les fonctionnalités disponibles et celles qui ne le sont pas selon le matériel utilisé.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=36

Ateliers de Brainstorming / Intelligence Artificielle / Choix du modèle

Complexité : 1 – Priorité : 3

L'IA nécessite une quantité considérable de données et de ressources de calcul pendant sa phase d'apprentissage et d'inférence, donc une consommation énergétique importante et une empreinte environnementale non négligeable.

Les impacts liés à la fabrication des microprocesseurs spécialisés, ainsi que les prévisions de consommation d'électricité et d'eau supplémentaires générées par l'IA, doivent amener à s'interroger en premier lieu sur la nécessité de recourir à l'IA pour répondre aux besoins des utilisateurs.

Afin de limiter l'impact environnemental des projets, il convient donc de s'interroger sur la pertinence de l'IA dans la réponse à apporter au besoin. Il s'agit d'évaluer si l'ajout d'un système d'IA apporte une réelle valeur ajoutée au service, tant pour les utilisateurs que pour l'efficacité globale du système. L'usage de l'IA doit être motivé par la confirmation du besoin via une méthode centrée sur l'utilisateur : « l'emploi d'une solution IA pour répondre à quel usage/finalité ? »

Moyens de mise en œuvre :

Il s'agit de ne pas intégrer de système d'IA tant que le besoin n'est pas clairement établi.

Cela commence par une identification des utilisateurs cibles et de leurs besoins spécifiques en lien avec le service numérique, avant de dresser une liste des fonctionnalités essentielles qui répondent à ces besoins.

Ultimement se pose alors la question de l'utilité d'un système d'IA : son ajout va-t-il réellement améliorer le service et répondre aux attentes des utilisateurs ? Est-il vraiment indispensable d'intégrer des systèmes ayant été entraîné ? Peut-on répondre au besoin en utilisant des algorithmes traditionnels ?

Si la solution IA est confirmée pour répondre au besoin, alors l'usage du service IA doit être optimisé dans le parcours de son utilisateur (au bon moment et dans le bon contexte), afin de garantir un usage raisonné des ressources pour répondre au juste besoin.

Critères de validation :

- Aucune phase d'entraînement n'est intégrée tant que le besoin n'est pas clairement démontré.
- La démonstration du besoin d'un système d'IA doit être fondée sur une analyse des publics cibles et des besoins associés, accompagnée de documents de référence tels que des études, des entretiens ou des persona qui ont permis de définir ces cibles.
- La déclaration d'écoconception doit justifier le lien entre les fonctionnalités du service et la nécessité d'un système d'IA, en mettant en avant la valeur ajoutée pour l'utilisateur. Elle doit justifier en quoi les solutions non IA ne permettent pas de répondre au besoin, même si elles sont adaptées ;

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=123
- Référentiel Général pour l'IA Frugale : BP01
- Référentiel Général pour l'IA Frugale : BP02

AB-ST-01 – Prévoir une stratégie de maintenance et de fin de vie du service numérique, et de décommissionnement des équipements

Ateliers de Brainstorming / Stratégie / Fin de vie du service
Complexité : 2 – Priorité : 3

Contre intuitivement, penser à la fin de vie de nos services dès leur conception est une bonne chose à faire. Cela inclut de penser la maintenance après l'arrêt des développements, mais aussi la fin de service. Pourquoi ? Parce que les environnements techniques qui ne sont plus utilisés continuent à consommer des ressources informatiques inutilement. Environ 25 % des serveurs physiques et virtuels sont alloués et tournent inutilement ! Arrêter un service, c'est aussi libérer des ressources physiques, supprimer les données inutiles, archiver les autres, et réaffecter les équipements.

Moyens de mise en œuvre :

- Définir une stratégie de maintenance et de décommissionnement : Mettre à jour régulièrement cette stratégie et fixer des dates de rappel. Si une partie du service ou le service entier n'est plus du tout utilisé, envisager de l'arrêter.
- Donner une nouvelle vie aux ressources libérées : En cas de décommissionnement, réutiliser, reconditionner ou recycler le matériel et les ressources libérées.
- Publication en open source : Si le service numérique est propriétaire, pourquoi pas prévoir de publier le code source en open source en cas de fin de vie pour maximiser la réutilisation potentielle du travail fourni lors de la phase de conception/développement par d'autres personnes.

Critères de validation :

- Une stratégie de maintenance et de décommissionnement est définie pour le service, incluant des dates de rappel pour les éléments non utilisés.
- Les actions prévues pour optimiser la seconde vie ou la fin de vie des ressources libérées en cas de décommissionnement sont documentées.
- Les résultats sont consignés dans la déclaration d'écoconception.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=45
- Les 115 bonnes pratiques d'écoconception web : 115

Rétrospective d'écoconception

Cérémonie régulière (à minima tous les 6 mois), spécifiquement dédiée à l'analyse des impacts des services et à l'étude des possibilités d'améliorer l'écoconception.

ECO-ST-01 – Réaliser régulièrement un bilan environnemental du service à l'aide d'une Analyse de Cycle de Vie

Rétrospective d'écoconception / Stratégie / Mesure
Complexité : 3 – Priorité : 3

Pour réduire l'empreinte environnementale du service numérique, il faut d'abord savoir l'estimer correctement. Cela implique d'avoir une vision globale et systémique des impacts environnementaux du service numérique, à chaque phase (début, usage, fin) et en intégrant les impacts environnementaux des équipements matériels utilisés, dans la production mais aussi dans l'usage de ce service numérique. Pour cela, la méthodologie de l'Analyse de Cycle de Vie est tout indiquée en tant que seule méthode capable de prendre en compte toutes les étapes d'un service sur plusieurs indicateurs environnementaux.

Moyens de mise en œuvre :

Mener une Analyse de Cycle de Vie (ACV) multicritère afin d'identifier les indicateurs permettant de documenter la majorité de l'empreinte environnementale du service (se référer au "Référentiel méthodologique d'évaluation environnementale des services numériques" de l'ADEME).

Les indicateurs d'impacts environnementaux à considérer prioritairement sont la consommation d'énergie primaire, les émissions de gaz à effet de serre (GES), la consommation d'eau et l'épuisement des ressources en métaux.

Fixer les objectifs de réduction de l'empreinte environnementale du service numérique (à court, moyen ou long terme) au regard du nombre d'utilisateurs.

Critères de validation :

- Des indicateurs multicritères ont été identifiés pour renseigner l'empreinte environnementale du service numérique.
- L'évaluation se base sur une méthodologie connue et détaillée
- Les limites de l'étude sont connus
- La base de données contenant les facteurs d'impact est consultable
- L'évaluation est périodique
- Des objectifs de réduction des impacts environnementaux sont fixés. Pour certains enjeux environnementaux comme le climat, les trajectoires devraient être alignées avec l'Accord de Paris (en se basant par exemple sur SBTi).

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=26

Rétrospective d'écoconception / Stratégie / Budget environnemental

Complexité : 3 – Priorité : 3

L'idée est de fixer des métriques techniques que le service doit respecter pour garantir un chargement rapide, peu importe les conditions d'accès (type de réseau, appareil utilisé). Tout comme un projet se fixe des contraintes budgétaires ou de délais, il est pertinent de se fixer des contraintes en matière de sollicitation de ressources.

Prenons l'exemple d'un site qui fournit des horaires de train. Les utilisateurs doivent pouvoir y accéder depuis un train, où la connexion peut être lente et instable via un smartphone. Dans ce contexte, si l'on estime qu'il est impératif que le site s'affiche en moins de 5 secondes en 3G pour éviter un taux de rebond ou d'abandons trop élevé, il est essentiel de déterminer des métriques limite comme le poids des ressources, leur nombre, leur ordre de chargement, etc... Des outils comme le site performance budget (<https://www.performancebudget.io/>) permettent de faire ces calculs rapidement.

En poussant la réflexion jusqu'au bout, on peut également fixer une durée maximale acceptable sur la durée de l'expérience utilisateur pour réaliser son action afin de réduire le temps d'utilisation du téléphone et la consommation de sa batterie, ainsi que le temps d'attention requis de l'utilisateur.

Moyens de mise en œuvre :

Les questions à se poser :

- Quelle est la durée de chargement maximale que mes utilisateurs peuvent tolérer ?
- Quelle est la durée de parcours maximale acceptable pour que mes utilisateurs accomplissent leur tâche (unité fonctionnelle) ?
- Quel est le seuil de taux de rebond que je peux accepter en raison d'un temps de chargement trop long ?
- Quelles sont les conditions de connexion les plus courantes lorsque mes utilisateurs accèdent à mon service ?
- Quelles sont les conditions de connexion moins favorables que certains de mes utilisateurs pourraient rencontrer en essayant d'accéder à mon service ?
- Quelle nouvelle audience pourrais-je atteindre si mon service était accessible dans des conditions de connexion moins optimales ?

Critères de validation :

- Les temps de chargement (Time to First Bit, Speed Index, Largest Contentful Paint) respectent la durée acceptable définie.
- Les utilisateurs parviennent à réaliser leurs tâches dans le temps imparti.
- Le taux de rebond est en dessous du seuil maximum acceptable.
- Les données d'utilisation réelles montrent que le service est utilisé dans des conditions de connexion variées.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=15>

ECO-ST-03 – Réaliser régulièrement des revues pour s’assurer du respect de la démarche d’écoconception

Rétrospective d’écoconception / Stratégie / Audit

Complexité : 2 – Priorité : 3

Les services numériques ne sont pas figés dans le temps, bien au contraire ils évoluent en permanence au gré des changements d’équipe, de l’ajout de contenu par les utilisateurs, de l’augmentation du trafic, des nouvelles fonctionnalités et l’introduction de nouvelles technologies. Pour garantir que la démarche d’écoconception soit maintenue sur le long terme et que l’empreinte environnementale du service n’augmente pas de façon démesurée, il nous faut effectuer des audits réguliers.

Ces audits permettent de s’assurer que les principes d’écoconception continuent à être appliqués, que les métriques techniques et le budget environnemental sont respectés, et favorisent également la transparence vis-à-vis des utilisateurs et des parties prenantes en publiant une déclaration d’écoconception qui reflète la performance environnementale du service tout en mettant à jours les données d’impact.

Moyens de mise en œuvre :

Planifier des audits réguliers de la performance environnementale et des pratiques sur le service en se basant sur le référentiel d’écoconception. La fréquence de ces évaluations doit être ajustée en fonction de la taille et de la nature du service numérique. Il est important de consigner les résultats de ces (auto)évaluations dans la déclaration d’écoconception, qui doit être mise à jour régulièrement, notamment à chaque changement significatif du service, par exemple lors de l’introduction de nouvelles fonctionnalités dimensionnantes ou de nouveaux services tiers. Cette déclaration doit également mentionner les actions envisagées pour améliorer la performance environnementale du service pour garder une dynamique d’amélioration continue.

Critères de validation :

- Une revue ou auto-évaluation régulière a été réalisée en appliquant le référentiel d’écoconception.
- Les résultats de cette évaluation sont documentés dans la déclaration d’écoconception du service.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=25

ECO-ST-04 – Identifier les opportunités d’allègement et supprimer les fonctionnalités non utilisées

Rétrospective d'écoconception / Stratégie / Allègement
Complexité : 2 – Priorité : 3

Lorsqu'on envisage de refondre un site web déjà existant, il est intéressant d'exploiter les données de navigation en provenance de solutions d'Analytics pour mieux comprendre le comportement des utilisateurs. Ces données pourront nous permettre de déceler des opportunités d'allègement. Les pages où les utilisateurs passent peu de temps ou, au contraire, celles où ils s'attardent trop longtemps peuvent nous indiquer des problèmes d'ergonomie ou de contenu. Les allers-retours fréquents entre deux pages peuvent signaler une confusion, ce qui ouvre la voie à une simplification du parcours. Les pages avec un taux de rebond élevé peuvent également révéler des attentes non satisfaites des utilisateurs, suggérant qu'une refonte ou une suppression de ces pages pourrait être bénéfique. Les pages les plus visitées évidemment nous indiquent là où les efforts d'écoconception peuvent avoir le plus d'impact. Les mots clés les plus recherchés par les utilisateurs peuvent également guider la restructuration du site pour fluidifier l'accès à l'information. La suppression des fonctionnalités peu ou plus utilisées (le fameux "Gras fonctionnel") permet de garder l'application légère en continuant de l'enrichir de nouvelles fonctionnalités.

Moyens de mise en œuvre :

Utiliser un outil d'Analytics pour mesurer l'utilisation réelle des fonctionnalités en production. Une fois les opportunités d'allègement identifiés, pour procéder à la suppression d'une fonctionnalité, deux approches peuvent être envisagées :

- la désactivation, qui consiste à utiliser un système de "feature flipping" pour empêcher son utilisation et son chargement via un flag
- la suppression, qui nécessite de supprimer le code associé tout en refactorisant le reste pour maintenir la cohérence de l'application.

Il convient également de poser la question des données associées à une fonctionnalité que l'on souhaite supprimer, quant au besoin de le conserver dans le système de stockage ou la base de données.

Critères de validation :

- Les données de navigation sont collectées de manière éthique et respectueuse de la vie privée.
- une analyse des comportements des utilisateurs a été réalisée.
- Les opportunités d'allègement des parcours et des fonctionnalités sont régulièrement considérés.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=14>
- Les 115 bonnes pratiques d'écoconception web : 3

Rétrospective d'écoconception / Spécifications / Accessibilité et Compatibilité

Complexité : 3 – Priorité : 3

Selon l'étude ADEME-Arcep, les terminaux représentent entre 65 et 92 % de l'empreinte environnementale du numérique, principalement en raison de leur fabrication. En rendant nos services compatibles avec des systèmes d'exploitation et des navigateurs web anciens, nous permettons à un plus grand nombre d'utilisateurs de continuer à utiliser leurs appareils sans avoir à les remplacer prématurément. Nous voulons avant tout garantir que nos services numériques restent utilisables le plus longtemps possible sur des appareils et logiciels anciens, afin de réduire l'obsolescence des terminaux de nos utilisateurs et prolonger au maximum leur durée de vie.

Moyens de mise en œuvre :

- Analyser les statistiques de fréquentation du service pour identifier les configurations techniques les plus courantes parmi nos utilisateurs. Attention au biais du survivant : les utilisateurs qui ne peuvent pas du tout utiliser le site n'apparaîtront pas dans ces statistiques.
- Assurer la compatibilité avec les versions anciennes des principaux navigateurs web.
- Si le service est une application native, il doit être utilisable sur les versions anciennes des systèmes d'exploitation encore en support long terme.
- Pour chaque fonctionnalité principale, assurez-vous que le service est compatible avec des équipements anciens.

Critères de validation :

- [Web] Les fonctionnalités critiques fonctionnent sur les principaux navigateurs web dans une version datée d'au moins deux ans, en prenant en compte la première date de mise à disposition en version stable.
- [Web] Les fonctionnalités critiques fonctionnent sur un PC portable ou un autre terminal équipé d'un microprocesseur mis sur le marché il y a dix ans ou plus.
- [Application native] Les fonctionnalités critiques fonctionnent sur les systèmes d'exploitation supportés ayant cinq ans, en prenant en compte la première date de première mise à disposition en version stable.
- [Application native] Les fonctionnalités critiques fonctionnent sur un équipement ancien (par exemple un smartphone, une tablette ou TV connectée) mis sur le marché il y a sept ans ou plus, dans la dernière version du système d'exploitation proposé par cet équipement.
- La déclaration d'écoconception du service numérique spécifie les versions minimales des systèmes d'exploitation supportés, les prérequis logiciels, les versions minimales des navigateurs web compatibles et leur année de sortie, ainsi que les caractéristiques matérielles et logicielles des équipements anciens permettant de faire fonctionner le service.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 57
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=37
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=40

Rétrospective d’écoconception / Spécifications / Accessibilité et Compatibilité

Complexité : 3 – Priorité : 3

L’accessibilité ne doit pas être perçue comme une contrainte, mais comme une opportunité d’améliorer l’expérience de tous les utilisateurs, y compris ceux en situation de handicap. En testant régulièrement l’accessibilité du service et en s’assurant que les bonnes pratiques sont appliquées, on garantit la qualité du service et sa conformité aux normes à venir.

Moyens de mise en œuvre :

Rappelons-nous d’abord que les outils d’évaluation ne remplacent pas l’expertise humaine, 80% des problèmes d’accessibilité ne peuvent pas être détectés par des outils automatisés.

Cependant, certains outils peuvent être utilisés pour accélérer l’identification des problèmes d’accessibilité qui n’auront pas à être remontés pendant un audit manuel. Par exemple :

- WAVE (Web Accessibility Évaluation Tool) : Outil permettant de vérifier certaines règles d’accessibilité et d’identifier les éléments problématiques sur une page web.
- Kastor.green : Outil permettant de vérifier certaines règles d’accessibilité du RGAA sur une page web.
- No Coffee : Extension qui aide à tester la lisibilité des couleurs, en simulant différentes déficiences visuelles.
- Colour Contrast Analyzer : Outil utile pour évaluer le contraste entre le texte et l’arrière-plan, garantissant ainsi une bonne lisibilité.

Et pour mener un audit complet :

- Kit d’audit RGAA : Ce kit fournit des outils et des modèles de documents pour réaliser un audit d’accessibilité.

Critères de validation :

- Les tests d’accessibilité sont réalisés à chaque étape clé du développement.
- Un audit d’accessibilité est réalisé au moins une fois par an pour s’assurer que le service reste conforme aux normes d’accessibilité.
- Des tests utilisateurs sont effectués avec des personnes en situation de handicap pour recueillir des retours directs sur l’accessibilité du service.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=59>

ECO-SP-03 – Assurer l’accessibilité du service numérique en bas débit et hors connexion

Rétrospective d’écoconception / Spécifications / Accessibilité et Compatibilité

Complexité : 2 – Priorité : 2

Il convient de s’assurer que votre service numérique reste accessible même pour ceux qui n’ont pas une connexion internet haut débit. On parle de gens qui peuvent être sur une connexion 3G ou même moins, en déplacement dans des zones blanches, ou qui n’ont pas accès à internet en permanence. Un service plus léger consomme moins de ressources réseau, et évite d’encourager les utilisateurs à se doter de services réseau plus performants (plus gros package de data ou accès à internet par satellite par exemple).

Moyens de mise en œuvre :

Tester votre service avec des connexions bas débit. Utilisez des outils comme lowband.com, webpagetest.org ou la console de développement de Firefox/Chrome pour simuler une connexion lente. Dans l’onglet "Network" ou "Réseau" des consoles navigateur, vous pouvez choisir le type de connexion (throttling) et mesurer les temps de chargement en les simulant. Si possible, servez vos contenus en qualité dégradée.

Pour les applications natives ou PWA (grâce aux ServiceWorker et au localStorage), pensez à prévoir un mode hors ligne pour les fonctionnalités qui peuvent fonctionner sans connexion.

Mesurez et améliorez le temps de réponse pour que tout reste fluide même en bas débit.

Critères de validation :

- Le service est utilisable avec une connexion bas débit (3G en mobilité et 512 Kbit/s en fixe).
- Le service est utilisable hors connexion pour les fonctionnalités clés qui peuvent techniquement fonctionner sans réseau.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=39
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=58>

Rétrospective d'écoconception / Spécifications / Optimisation et Évolution

Complexité : 2 – Priorité : 2

Créer un service écoconçu ne suffit pas ; il est tout aussi important de s'assurer qu'il continuera à respecter les principes d'écoconception au fil du temps, des évolutions et de l'introduction de nouvelles fonctionnalités. Pour cela, il est nécessaire de laisser une documentation exhaustive à destination des futurs équipes qui explique les spécificités du service, les bonnes pratiques à suivre, ainsi que les connaissances nécessaires pour maintenir et faire évoluer le contenu sur la lignée des choix d'écoconception qui ont été pris lors de sa création.

Moyens de mise en œuvre :

Quelques questions à se poser :

- Qui sera chargé de mettre à jour et gérer le contenu du service ? Comment lui transmettre les bonnes pratiques concernant la gestion du contenu ?
- Quelles sont les bonnes pratiques et les connaissances essentielles à transmettre aux futures équipes de développement ?
- Quels types de documents peuvent être créés pour faciliter la compréhension des choix d'écoconception de l'équipe initiale ?

Critères de validation :

- Des documents détaillés et accessibles ont été créés, incluant des guides de pratiques et des listes de choix de conception.
- Un processus de mise à jour des documents a été établi, avec des responsables clairement identifiés.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=60>

Rétrospective d'écoconception / Architecture / Mises à jour

Complexité : 2 – Priorité : 2

Il est conseillé de permettre aux utilisateurs de choisir le type de mises à jour qu'ils souhaitent installer. Les mises à jour se divisent en deux catégories principales : les mises à jour correctives (ou de conformité), qui corrigent des failles de sécurité ou des bugs, et les mises à jour évolutives, qui ajoutent ou modifient des fonctionnalités. Dans le cas des applications mobiles en grande partie, l'ajout de nouvelles fonctionnalités va ultimement déboucher sur une incompatibilité des équipements devenus trop vieux.

En informant les utilisateurs du contenu et de l'impact potentiel de chaque mise à jour et en leur laissant le choix de ne pas installer les mises à jour évolutives, on leur permet de prolonger la durée de vie de leurs appareils. Cette approche aide également à éviter l'obsolescence psychologique en limitant les ajouts de fonctionnalités non essentielles qui pourraient nécessiter des ressources matérielles supplémentaires.

Moyens de mise en œuvre :

Installer de façon dissociée les mises à jour correctives, ou toute autre mise à jour essentielle à la conformité et à la sécurité du service numérique ou du terminal de l'utilisateur, et les mises à jour évolutives qui ne sont pas nécessaires à la conformité du bien.

De façon générale, les mises à jour évolutives ne doivent pas empêcher le service numérique de fonctionner sur toute la durée de maintenance des systèmes d'exploitation supportés (sous réserve d'absence de contraintes techniques ou de sécurité).

Critères de validation :

- La description des mises à jour d'une application (changelog) indique clairement s'il s'agit d'une mise à jour de sécurité/maintenance (« corrective ») ou d'une mise à jour évolutive ajoutant des fonctionnalités.
- L'utilisateur a le choix d'installer uniquement les mises à jour essentielles à la conformité et à la sécurité du service numérique ou du terminal de l'utilisateur et refuser les mises à jour évolutives.
- Les mises à jour évolutives non essentielles à la conformité du service n'empêchent pas le service de fonctionner pendant toute la durée de maintenance des systèmes d'exploitation supportés.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=21>
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=56

Ateliers UX/UI

Ateliers au cours desquelles l'équipe Scrum, en collaboration avec les designers UX/UI, explore, conçoit et affine l'expérience utilisateur d'une fonctionnalité ou d'un produit, en s'appuyant sur les spécifications du produit et les retours utilisateurs.

Ateliers UX/UI / Contenus / Accessibilité et Inclusion

Complexité : 3 – Priorité : 3

L'objectif est de rendre les textes sur le web à la fois accessibles et faciles à lire pour tous les utilisateurs et sur tous les supports. Pour ce faire, il est essentiel d'écrire des textes courts et directs, en privilégiant la concision. Si le contenu est long ou complexe, il est préférable de le diviser en plusieurs pages pour éviter de surcharger le lecteur.

Moyens de mise en œuvre :

- Si un sujet nécessite une longue explication, le découper en plusieurs pages ou le traiter dans un document à télécharger. Cela permet aux utilisateurs de n'afficher que les pages qu'ils souhaitent lire et de ne télécharger que les fichiers qui les intéressent.
- Assurez-vous que l'information reste visible et compréhensible même lorsque les feuilles de style sont désactivées ou que la taille des caractères est augmentée à 200%. Ne mettez pas d'information essentielle dans les images et évitez de justifier le texte. Utilisez un interlignage d'au moins 1,5 dans les paragraphes et un espacement entre les paragraphes d'au moins 1,5 fois plus grand que la valeur de l'interligne pour respecter les normes AA d'accessibilité.
- Rédigez de manière concise et privilégiez les listes à puces plutôt que les blocs de texte. Expliquez toutes les abréviations, sigles et acronymes. Pour répondre aux exigences AAA d'accessibilité, le contenu doit être compréhensible pour toute personne ayant un niveau de lecture équivalent à la fin de l'école primaire ou moins. Sinon, proposez une façon alternative de présenter l'information afin qu'elle soit compréhensible pour ce type de profils.

Critères de validation :

- Tous les textes affichés sur le site web obtiennent une note issue de l'indice de lisibilité Flesch-Kincaid d'au moins 60.
- L'information reste visible et compréhensible même lorsque les feuilles de style sont désactivées ou que la taille des caractères est augmentée à 200%.
- Les textes ne contiennent pas d'information essentielle dans les images et ne sont pas justifiés.
- L'interlignage dans les paragraphes est d'au moins 1,5 et l'espacement entre les paragraphes est d'au moins 1,5 fois plus grand que la valeur de l'interligne.
- Toutes les abréviations, sigles et acronymes sont expliqués.

Liens :

- <https://accessibilite.numerique.gouv.fr/>
- Les 115 bonnes pratiques d'écoconception web : 110
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=48>

Ateliers UX/UI / Contenus / Accessibilité et Inclusion

Complexité : 2 – Priorité : 3

L'idée ici, c'est de rendre votre contenu accessible à tout le monde, peu importe leurs capacités ou leur connexion internet. Le texte, même stylisé avec du HTML/CSS, consomme beaucoup moins de bande passante que des vidéos ou des fichiers audios. En offrant une alternative textuelle, vous permettez à vos utilisateurs de lire plutôt que de regarder ou d'écouter, ce qui peut être une solution de repli pour ceux qui ont une connexion lente ou des limitations de données. C'est aussi un gros plus pour l'accessibilité : les malentendants peuvent lire le contenu, et les malvoyants peuvent utiliser des lecteurs d'écran si le texte inclut une description des éléments visuels.

De même, le texte directement inclus dans les images n'est pas accessible parce que les lecteurs d'écran ne peuvent pas le lire. Il s'agit alors de ne pas intégrer de texte important dans l'image elle-même, mais en overview dans un bloc HTML puis dans la description alt-text. Cette description est lue par les lecteurs d'écran et s'affiche aussi quand la connexion est trop lente pour charger les images. C'est une obligation du RGAA (Référentiel Général d'Accessibilité pour les Administrations).

Moyens de mise en œuvre :

- Fournir une alternative textuelle aux vidéos et audios. Une vidéo de 30 minutes peut peser environ 500 Mo, un podcast de la même durée environ 30 Mo, et l'équivalent en texte moins d'1 Mo.
- Pour les images, utilisez la balise alt-text. Cette balise doit contenir une description claire et concise de l'image, incluant l'objet, l'action et le contexte. Par exemple, pour une photo d'une pancarte "Sauvez les baleines" dans une foule, vous pourriez écrire : "Pancarte 'Sauvez les baleines' dans une foule." Suivez les conseils de guides comme "How to write an image description" pour vous assurer que vos descriptions sont compréhensibles.

Critères de validation :

- Toutes Les images ont une description alt-text appropriée et conforme aux recommandations du RGAA.
- Les médias audios et vidéos ont tous une alternative textuelle.

Liens :

- <https://accessibilite.numerique.gouv.fr/>
- Les 115 bonnes pratiques d'écoconception web : 30
 - <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=35>
 - <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=36>
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=68

UX-CO-03 – Opter pour les choix les plus sobres entre le texte, l’image, l’audio ou la vidéo

Ateliers UX/UI / Contenus / Expérience Utilisateur

Complexité : 1 – Priorité : 1

Minimiser la sollicitation de ressources en choisissant des formats de contenu qui répondent aux besoins des utilisateurs tout en étant les plus légers possibles. Par exemple, une vidéo, même optimisée, consomme généralement beaucoup plus de ressources qu’un simple texte accompagné d’images.

Moyens de mise en œuvre :

Questionner la nécessité d’utiliser un média en particulier, qu’il s’agisse d’une vidéo, d’une animation ou d’un enregistrement audio. Il convient de documenter ces réflexions pour justifier les choix effectués.

En règle générale, il est bon de garder à l’esprit que :

- Un texte est généralement moins lourd qu’une image.
- Une image pèse moins qu’un fichier audio.
- Un fichier audio est moins lourd qu’une vidéo.

Ainsi, lorsque le besoin d’un média est avéré, il est préférable de choisir la solution la plus sobre qui répondra tout de même aux attentes et aux besoins des utilisateurs.

Critères de validation :

- Le service n’utilise pas de contenu vidéo, audio ou animé, ou
- L’utilisation de vidéo, d’audio ou d’animation a été justifiée par le choix de la solution la plus sobre disponible, en tenant compte des besoins essentiels des utilisateurs et des fonctionnalités du service.
- Les choix effectués doivent être clairement justifiés dans la déclaration d’écoconception.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=69

UX-CO-04 – Bannir les fonds vidéo et la lecture automatique des contenus multimédias

Ateliers UX/UI / Contenus / Optimisation Technique

Complexité : 1 – Priorité : 3

Plus un utilisateur passe de temps sur un site ou une application, plus l’empreinte environnementale associée à cet usage sera élevée, les ressources réseaux et systèmes sur le terminal étant mobilisées plus longtemps. S’il appartient bien entendu à l’utilisateur de décider quel temps il souhaite consacrer à un usage, les pratiques de développement du site ou de l’application concernée peuvent avoir un impact direct sur cette durée d’utilisation par l’exploitation de pratiques de captation de l’attention. Le déclenchement de contenus et leur préchargement sans consentement de l’utilisateur doivent donc être évités. L’article 19 de la proposition de loi du Sénat prévoyait l’interdiction de la lecture automatique des vidéos. Bien que cette mesure n’ait pas été rendue obligatoire dans l’immédiat, on peut supposer que cette interdiction advienne dans les années à venir. Enfin, l’activation automatique du son dans une vidéo peut gêner les personnes malvoyantes utilisant un lecteur d’écran, voire peut les empêcher de naviguer sur le site. Dans tous les cas, il faut toujours permettre aux utilisateurs de régler le volume audio, d’éviter les arrière-plans sonores ou bien de permettre de les désactiver.

Moyens de mise en œuvre :

Désactiver par défaut le chargement et la lecture automatiques de contenus vidéo et/ou sonores dans les paramètres du service. Si ce n’est pas possible, permettre à l’utilisateur de supprimer ces fonctionnalités via une interface directement accessible et visible. Autant que possible, ne pas utiliser de graphismes animés non contrôlables par l’utilisateur (images gif animées notamment). Les fonds vidéo et/ou audio en lecture automatique devront en particulier être évités, s’ils sont à visée purement esthétique. Pour les animations jugées essentielles vis-à-vis des fonctionnalités du service, proposer une possibilité à l’utilisateur de mettre en pause ces éléments. Lorsque l’animation visuelle a une durée de plus de 4 secondes ou qu’un son a une durée de plus de 2 secondes, doter systématiquement l’objet multimédia des moyens de contrôle nécessaires : démarrage, arrêt, muet ou volume. Permettre de les désactiver simplement dès lors que ces contenus dépassent 4 secondes et minimiser leur taille.

Critères de validation :

- Le service n’inclut pas de lecture automatique de contenu par défaut.
- Si ce n’est pas possible, le service numérique donne la possibilité à l’utilisateur de supprimer facilement le chargement ou le lancement automatique de vidéos ou contenus audios.
- Le recours à des animations visuelles, clignotements ou défilements automatiques non contrôlables est limité.
- Les animations non contrôlables jugées essentielles vis-à-vis des fonctionnalités du service peuvent être mises en pause par l’utilisateur dès qu’elles dépassent 4 secondes pour l’animation visuelle et 2 secondes pour l’audio.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=61
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=37>

UX-CO-05 – Éviter d'intégrer des plugins de lecteur vidéo

Ateliers UX/UI / Contenus / Optimisation Technique

Complexité : 1 – Priorité : 2

Si une vidéo n'est pas essentielle à la navigation ou à l'expérience utilisateur, comme c'est souvent le cas pour des contenus tels que des interviews ou des présentations de produits, il est préférable d'opter pour une solution plus légère. Par exemple, remplacer la vidéo par une image cliquable qui lancera le chargement d'un module ou redirigera l'utilisateur vers une plateforme d'Hébergement de vidéo dédiée permet non seulement d'alléger le chargement de la page, mais aussi d'améliorer l'expérience utilisateur en évitant des temps d'attente inutiles.

Moyens de mise en œuvre :

Il existe plusieurs alternatives intéressantes. Pensez à utiliser des plateformes comme PeerTube pour héberger vos vidéos, ce qui vous permet de bénéficier d'une solution décentralisée et moins lourde que celles des GAFAM. Une autre option consiste à intégrer un lecteur HTML5. Ce type de lecteur est particulièrement avantageux car il est nativement supporté par le navigateur et ne va pas nécessiter de chargement de modules supplémentaires, la vidéo ne sera réellement chargée que lorsque l'utilisateur décidera de cliquer sur le bouton "play". Cela réduit considérablement le poids de la page au chargement initial et améliore la réactivité du site.

Critères de validation :

- Les vidéos non essentielles sont remplacées par des images cliquables.
- Un lecteur HTML5 est utilisé pour les vidéos essentielles, avec un chargement à la demande.
- Les vidéos sont hébergées sur des plateformes décentralisées comme PeerTube.
- Les temps de chargement de la page sont mesurés et montrent une amélioration par rapport à une intégration classique de vidéos.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=39>

Ateliers UX/UI / Contenus / Optimisation Technique

Complexité : 2 – Priorité : 3

Lorsque vous envisagez d'intégrer des vidéos ou des fichiers audios dans votre service, il est essentiel de garder à l'esprit l'impact que cela peut avoir sur l'expérience utilisateur et la sollicitation des ressources matérielles. Heureusement, il existe des outils de compression qui permettent de réduire la taille des fichiers sans sacrifier la qualité perçue. Par exemple, avec un logiciel comme Handbrake, vous pouvez compresser une vidéo HD d'une minute et trente-cinq secondes en 720p, ce qui peut réduire son poids de 38 %, passant ainsi de 11,7 Mo à une taille plus gérable.

Moyens de mise en œuvre :

Choisir judicieusement le niveau de compression en fonction de l'utilisation prévue. Par exemple, si vous publiez sur les réseaux sociaux, une vidéo en 480p, plus légère, peut-être plus appropriée qu'une vidéo en 720p.

Pour les fichiers audios, des outils comme Adobe Audition ou l'encodeur gratuit Lame peuvent être utilisés pour compresser les bandes sonores en mp3, sans que l'utilisateur final ne remarque une perte de qualité significative.

Critères de validation :

- Les vidéos et fichiers audios sont compressés à un niveau approprié pour leur contexte d'utilisation.
- La taille des fichiers est significativement réduite par rapport à leur taille originale sans perte de qualité perceptible.
- Les utilisateurs finaux ne signalent pas de dégradation de la qualité lors de la lecture des vidéos ou des fichiers audios.
- Les temps de chargement des vidéos et des fichiers audios sont améliorés, en particulier sur les connexions mobiles.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=38>

Ateliers UX/UI / Contenus / Expérience Utilisateur

Complexité : 3 – Priorité : 3

Quand un service propose beaucoup de contenus, un travail important doit être mené sur la façon de l'explorer et le trouver. Il est recommandé d'offrir aux utilisateurs les trois méthodes de navigation suivante : un menu de navigation, un plan du site et un moteur de recherche. Ces éléments sont cruciaux pour aider les visiteurs à trouver rapidement ce qu'ils cherchent. De plus, il est important de décrire le contenu de chaque page dans les métadonnées, ce qui contribue à améliorer la visibilité et la pertinence des moteurs de recherche.

Moyens de mise en œuvre :

Commencez par définir une structure claire pour votre contenu. Intégrez les 3 méthodes de navigation : Menu, plan, moteur de recherche.

Pour la recherche, il est recommandé de proposer une méthode de recherche fulltext prenant en charge les fautes de frappes, les synonymes et la traduction. Un outil open source comme Meiliseach permet de créer facilement un endpoint permettant toutes ces fonctions.

Assurez-vous que chaque page contient des métadonnées bien rédigées, décrivant le contenu de manière concise et pertinente. Pensez également à tester la navigation avec des utilisateurs pour identifier d'éventuels points de friction.

Critères de validation :

- Les feuilles de style sont utilisées pour structurer le contenu avec des titres et sous-titres clairs.
- Au moins deux méthodes de navigation parmi le menu de navigation, le plan du site et le moteur de recherche sont mises en place et fonctionnelles.
- Les métadonnées de chaque page décrivent correctement le contenu et sont optimisées pour les moteurs de recherche.
- Des tests utilisateurs ont été réalisés pour s'assurer que la navigation est intuitive et efficace.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=48>

UX-CO-08 – Opter pour les options média par défaut les plus sobres pour le téléchargement et la lecture

Ateliers UX/UI / Contenus / Expérience Utilisateur

Complexité : 1 – Priorité : 3

Les options par défaut sont souvent celles que les utilisateurs choisissent sans y réfléchir, ce qui en fait des leviers puissants pour influencer leur comportement. En prenant le temps de bien sélectionner ces paramètres, on peut réduire significativement la sollicitation de ressources, notamment réseau. Par exemple, en évitant le téléchargement automatique de médias dans la plus haute définition possible, on peut diminuer significativement la quantité de données transférés par notre service.

Moyens de mise en œuvre :

- Désactiver le téléchargement automatique des médias lorsque l'utilisateur utilise des données mobiles.
- Permettre aux utilisateurs de choisir les types de médias à télécharger/lancer automatiquement, par exemple en excluant les vidéos.
- Ajuster l'option par défaut pour proposer un équilibre entre sollicitation des ressources et recherche de qualité. Par exemple, un bouton pour télécharger une image pourra par défaut lancer le téléchargement en version HD plutôt que 4K jusqu'à ce que l'utilisateur change cette option dans son profil ou via une sélection.

Critères de validation :

- Les paramètres par défaut ont été modifiés pour désactiver le téléchargement automatique des médias sur les données mobiles.
- Les utilisateurs ont la possibilité de personnaliser les types de médias à télécharger/lancer automatiquement.
- Des informations sont fournies aux utilisateurs concernant les implications de ces choix sur la consommation de ressources (poids des contenus affichés par exemple).

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=55>

Ateliers UX/UI / Contenus / Expérience Utilisateur

Complexité : 1 – Priorité : 2

Lorsqu'un document est particulièrement long (et lourd), il peut être judicieux de proposer un aperçu ou un résumé directement sur la page avant de devoir le télécharger. Cela permet aux utilisateurs de décider s'ils souhaitent vraiment télécharger le document complet, évitant ainsi des téléchargements inutiles qui sollicitent le réseau.

Moyens de mise en œuvre :

- Créer un résumé concis du document, sous forme d'infographie ou d'abstract, qui met en avant les points clés et les informations les plus pertinentes.
- Proposer deux formats de documents : une version courte et une version longue, en veillant à ce que la version courte soit suffisamment informative pour répondre aux besoins des utilisateurs sans nécessiter le téléchargement de la version complète.
- Indiquer clairement le poids de chaque document à côté des liens de téléchargement, afin que les utilisateurs puissent évaluer la taille des fichiers avant de les télécharger.
- Tester l'interface pour s'assurer que le résumé est bien visible et que les utilisateurs comprennent facilement les différences entre les deux versions.

Critères de validation :

- Un résumé est présent et accessible sur la page du document.
- Les deux versions du document (courte et longue) sont disponibles au téléchargement.
- Le poids de chaque version est clairement indiqué à côté des liens de téléchargement.
- Des retours d'utilisateurs montrent qu'ils trouvent le résumé utile pour prendre leur décision de téléchargement.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=54>

Ateliers UX/UI / Contenus / Référencement et Visibilité

Complexité : 2 – Priorité : 2

Un bon référencement minimise le temps passé à naviguer sur le site et par conséquent réduit le nombre de données transférées et la charge serveur. En optimisant le site pour les moteurs de recherche, on facilite la tâche à l'utilisateur, qui peut ainsi trouver ce qu'il cherche sans avoir à parcourir l'arborescence du site. Cela contribue à une sollicitation de ressources moindre lors de son parcours.

En intégrant des pratiques d'écoconception, on allège le poids des pages, ce qui se traduit par des temps de chargement plus rapides. Ces temps de chargement sont un critère important pour les moteurs de recherche, qui les prennent en compte pour classer les résultats.

En parallèle, de nombreuses bonnes pratiques d'écoconception s'alignent avec les exigences d'accessibilité, un autre facteur qui peut booster le référencement d'un site.

Un site conçu dans une démarche d'écoconception a de meilleures chances de se positionner favorablement dans les résultats de recherche, car il allie optimisation pour le référencement et respect des principes de sobriété et d'accessibilité.

Moyens de mise en œuvre :

- Effectuer un audit de référencement pour identifier les points d'amélioration.
- Optimiser les balises HTML (titres, descriptions, balises alt pour les images) pour qu'elles soient pertinentes et concises.
- Réduire le poids des fichiers (images, vidéos, scripts) en utilisant des formats adaptés et en compressant les fichiers.
- Mettre en place une structure de liens internes logique pour faciliter la navigation et l'indexation par les moteurs de recherche.
- Utiliser des mots-clés pertinents et recherchés dans le contenu tout en veillant à ce qu'ils soient intégrés de manière naturelle.
- S'assurer que le site est responsive et s'affiche correctement sur tous les types d'appareils.
- Tester régulièrement la vitesse de chargement des pages et apporter des ajustements si nécessaire.

Critères de validation :

- On peut accéder directement à un contenu via les résultats de moteurs de recherche comme Google.
- Le site obtient un score de performance satisfaisant sur des outils comme Google PageSpeed Insights et Google Lighthouse.
- Le temps de chargement des pages est inférieur à 3 secondes.
- Le site est accessible sur différents appareils et navigateurs sans perte de fonctionnalité
- Les résultats de recherche montrent une amélioration du classement pour les mots-clés ciblés.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=48>

UX-FE-01 – Informer l'utilisateur qu'un traitement est en cours en arrière-plan

Ateliers UX/UI / Frontend / Expérience Utilisateur

Complexité : 1 – Priorité : 1

Pour éviter des requêtes simultanées qui pourraient surcharger le système ou créer des confusions, il faut indiquer à l'utilisateur qu'un traitement est en cours en arrière-plan. Ce faisant, on lui évite de cliquer plusieurs fois sur le même bouton, ce qui pourrait entraîner des erreurs ou des comportements inattendus en plus d'envoyer plusieurs requêtes simultanément à travers le réseau.

Moyens de mise en œuvre :

Désactiver le bouton ou l'élément qui a déclenché l'action dès que l'utilisateur la déclenche. Par exemple, si un utilisateur soumet un formulaire, le bouton de soumission doit devenir inactif pour éviter qu'il ne soit cliqué à nouveau pendant que le traitement est en cours. En parallèle, afficher un indicateur visuel, comme un spinner ou un message d'attente, qui informe l'utilisateur que son action est en cours de traitement. Il peut également être utile de fournir une estimation du temps que cela pourrait prendre, afin de gérer les attentes de l'utilisateur.

Critères de validation :

- Lorsqu'une action est initiée, le bouton ou l'élément déclencheur devient inactif.
- Un indicateur visuel ou un message d'attente est affiché pour signaler que le traitement est en cours.
- L'utilisateur reçoit, si possible, une indication de la durée approximative du traitement.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=104

Ateliers UX/UI / Spécifications / Mobile First

Complexité : 3 – Priorité : 3

Lorsque l'on commence par concevoir pour les écrans d'ordinateurs ("desktop"), on a souvent tendance à ajouter énormément de contenu pour remplir les espaces vides. L'adaptation en version mobile représente ensuite un défi pour tout intégrer dans un petit écran. L'approche "Mobile first" consiste à concevoir d'abord son service numérique pour les appareils mobiles. Cela permet de se concentrer sur l'essentiel, de réduire les fonctionnalités et contenus accessoires, et de s'assurer que notre service fonctionne bien sur les appareils mobiles, utilisés par environ 55% des internautes (Statista, 2021). Cette approche "mobile first" doit être pensée pour des terminaux mobiles peu puissants et avec une connexion réseau faible (3G plutôt que 4G). Ces contraintes techniques obligent les concepteurs à se concentrer sur l'essentiel et à produire un service numérique sobre.

Moyens de mise en œuvre :

- Les questions à se poser :
 - Qu'est-ce qui est essentiel ?
 - Quel est le minimum de technologie nécessaire pour résoudre le problème ?
 - Est-ce que cela fonctionne sur mobile et desktop ?
 - Est-ce que la taille des boutons et champs est adaptée au mobile ?
- Côté serveur, on peut utiliser les client hints, ou à défaut l'identifiant du navigateur couplé à un tableau de capacités des navigateurs (user agent sniffing) si l'on veut adapter la quantité de données renvoyés au terminal.
- Côté client, les media queries (notamment dans les attributs media des <link> pour la sélection de feuilles de styles CSS), les attributs srcset et sizes des , les sous-éléments <source> des <picture>, <video> et <audio> pourront être utiles. Les mêmes informations peuvent être récupérées par des API JavaScript pour éventuellement charger du code et/ou du contenu complémentaire dynamiquement.

Critères de validation :

- Le service a été conçu avec une approche "mobile first".
- Les performances du site/service sur des terminaux mobiles peu puissants et avec une connexion 3G sont satisfaisantes.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=19>
- Les 115 bonnes pratiques d'écoconception web : 4
- <https://www.apl-datacenter.com/wp-content/uploads/2020/02/Greenconcept-Livre-Blanc-VF.pdf>

UX-ST-01 – Définir les cibles utilisatrices, les besoins métiers et les attentes réelles des utilisateurs-cibles

Ateliers UX/UI / Stratégie / Persona
Complexité : 1 – Priorité : 3

Une connaissance approfondie des cibles permet d'éviter de surcharger le service avec des fonctionnalités superflues ou, à l'inverse, de le rendre trop limité pour répondre aux attentes. Cela implique de comprendre les habitudes, les besoins spécifiques et les comportements de nos utilisateurs.

Les incertitudes quant aux usages peuvent mener à des extrapolations erronées des besoins, souvent au-delà de ce que les utilisateurs attendent réellement.

Moyens de mise en œuvre :

Les outils et techniques issus de la recherche en expérience utilisateur (UX research) permettent de bien définir les cibles utilisatrices. Cela peut inclure :

- La réalisation d'études concurrentielles pour comprendre le paysage existant.
- L'analyse des services déjà en place pour identifier les points forts et les lacunes.
- La création de persona, afin de mieux cerner les besoins et comportements.
- La conduite d'entretiens ou de sondages avec les utilisateurs pour recueillir des informations directes sur leurs attentes.
- L'observation des utilisateurs en situation réelle pour comprendre leurs interactions avec le service.

Pour identifier les besoins métiers et les attentes des utilisateurs, plusieurs étapes peuvent être suivies :

- Organiser des entretiens avec les parties prenantes et les métiers concernés pour recueillir leurs perspectives.
- Mener des recherches UX ciblées auprès des utilisateurs identifiés.
- Définir clairement qui sont les utilisateurs primaires et secondaires.
- Analyser les statistiques d'usage si un service existe déjà, afin de mieux comprendre comment il est utilisé.

Critères de validation :

- Les documents de référence issus de la phase de recherche sont accessibles, incluant les entretiens utilisateurs, les études UX, les benchmarks, les persona et les études marketing.
- Les choix effectués en matière de conception, basés sur cette compréhension des utilisateurs, sont bien documentés et justifiés.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=22

UX-UX-01 – Optimiser le parcours utilisateur pour une expérience fluide

Ateliers UX/UI / UX/UI / Expérience Utilisateur

Complexité : 3 – Priorité : 3

Il s'agit de réduire le temps que l'utilisateur passe sur le service numérique en optimisant les fonctionnalités principales et en rendant le parcours utilisateur aussi fluide et rapide que possible. Cela passe par la simplification des étapes, la suppression des éléments inutiles et l'amélioration des temps de réponse. Une expérience utilisateur bien conçue non seulement améliore la satisfaction des utilisateurs, mais contribue également à réduire la sollicitation de ressources.

Moyens de mise en œuvre :

- Phase de conception : Commencer par éliminer les fonctionnalités non essentielles. Se concentrer sur les unités fonctionnelles principales du service numérique, comme "réserver un billet", "rechercher un terme", "trouver une adresse", etc.
- Analyse UX : Utiliser des outils UX pour comprendre les usages des utilisateurs, via des sondages, des interviews, des enquêtes utilisateurs et des tests-U.
- Optimisation : Si le service existe déjà, mettre en place un système d'analyse non intrusif et respectueux de la vie privée pour suivre les parcours de navigation. Analyser régulièrement ces statistiques pour améliorer l'expérience utilisateur en réduisant les points de friction.

Critères de validation :

- Les parcours de navigation sont optimisés et recentrés autour des fonctionnalités essentielles, d'après les outils UX et les statistiques d'usage.
- Des indicateurs techniques pour les parcours identifiés ont été mis en place pour assurer l'optimisation continue du parcours de navigation (Total Blocking Time, Largest Contentful Paint, ...).
- Aucun points de friction (situations ou interactions qui contribuent à dégrader l'expérience utilisateur et à ralentir le parcours) n'est identifié à la suite d'interview utilisateurs.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=64
- Les 115 bonnes pratiques d'écoconception web : 5
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=17>

UX-UX-02 – Indiquer à l'utilisateur que l'utilisation d'une fonctionnalité implique des impacts environnementaux importants

Ateliers UX/UI / UX/UI / Expérience Utilisateur
Complexité : 1 – Priorité : 2

L'objectif ici est de rendre visibles pour l'utilisateur les conséquences environnementales des actions qui consomment le plus de ressources. Il s'agit d'informer les utilisateurs avant qu'ils n'utilisent une fonctionnalité, surtout si celle-ci est plus gourmande en ressources par rapport aux autres aspects du service, par exemple des fonctionnalités liées à l'utilisation de modèles de langage génératifs.

Moyens de mise en œuvre :

Pour chaque fonctionnalité du service qui pourrait avoir un impact environnemental notable, affichez une information préalable à l'utilisateur et laissez le choix à l'utilisateur d'effectuer l'action.

Par exemple, lors de la saisie d'un texte en 2 langues, nous pouvons imaginer laisser le choix à l'utilisateur de faire appel à un modèle de langage spécialisé dans la traduction via une action manuelle clairement identifiée comme étant impactante en termes de ressources, ou le laisser écrire lui-même le second texte sans utiliser la traduction.

Si possible, il est recommandé d'inclure des informations sur les équivalences en termes d'impact environnemental, tout en précisant la source et la méthodologie utilisées, et en adoptant une approche multicritère qui ne se limite pas uniquement à l'équivalent CO2. Par exemple, "l'appel à notre service de traduction automatique consomme l'équivalent de 0,5L d'eau par requêtes".

Critères de validation :

- Les fonctionnalités ayant un impact environnemental significatif sont clairement identifiées et signalées à l'utilisateur.
- L'utilisateur a le choix de ne pas utiliser les fonctionnalités ayant un impact environnemental significatif

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=74

UX-UX-03 – Fournir à l'utilisateur un moyen de contrôle sur ses usages

Ateliers UX/UI / UX/UI / Expérience Utilisateur

Complexité : 2 – Priorité : 2

L'idée ici est de donner à l'utilisateur les moyens de prendre en main son usage du service numérique, afin de réduire les sollicitations de ressources qui en découlent. En permettant à l'utilisateur de mieux comprendre et contrôler son utilisation, on vise à encourager des comportements plus responsables et à minimiser sa consommation. Cela passe par une sensibilisation et une autonomie accrue, permettant à chacun de faire des choix éclairés concernant ses usages numériques.

Moyens de mise en œuvre :

Pour mettre cela en pratique, le service numérique doit fournir des informations sur le temps passé à l'utiliser ainsi que sur les sollicitations de ressources qui en résultent. Ces informations doivent être basées sur des normes et standards reconnus pour garantir leur fiabilité, comme les « Product Category Rules » élaborées par l'ADEME spécifiquement pour les services numériques. De plus, il est essentiel que l'utilisateur ait la possibilité de choisir, par défaut ou à sa demande, des modes d'affichage et d'utilisation « sobres ». Ces modes doivent être conçus pour réduire la consommation de ressources et d'énergie.

Critères de validation :

- Le service fournit des données sur le temps d'utilisation et les sollicitations de ressources associées, en utilisant des normes reconnues.
- L'utilisateur a accès à des options d'affichage et d'utilisation « sobres » qui sont facilement activables.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=77

UX-UX-04 – Permettre à l'utilisateur de décider de l'activation d'un service tiers

Ateliers UX/UI / UX/UI / Expérience Utilisateur

Complexité : 1 – Priorité : 2

En laissant à l'utilisateur la possibilité de choisir s'il souhaite ou non activer des services tiers non critiques, on réduit la quantité de ressources sollicitées par le service. L'intégration forcée et par défaut de chatbot AI à des outils de communication est un très mauvais exemple à ne surtout pas suivre.

Moyens de mise en œuvre :

S'assurer que, sauf si un service tiers est critique pour le fonctionnement de base du service, celui-ci ne soit chargé qu'à la demande explicite de l'utilisateur. Cela implique de concevoir l'interface de manière que l'utilisateur soit clairement informé des services tiers disponibles et puisse choisir de les activer ou non. De plus, si ces services nécessitent le traitement de données personnelles, il est impératif de respecter les exigences du RGPD, qui stipule que le consentement de l'utilisateur doit être obtenu avant tout traitement de données.

Critères de validation :

- L'activation des services tiers intégrés au service numérique est soumise à un consentement clair et explicite de l'utilisateur.
- L'utilisateur est informé de manière précise sur les implications de l'activation de ces services, notamment en ce qui concerne le traitement de ses données personnelles.
- Une information spécifique est fournie concernant la sollicitation de ressources associée à l'activation de ces services, permettant à l'utilisateur de prendre une décision éclairée.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=66

UX-UX-05 – Éviter le recours à des procédés manipulatoires dans l'interface utilisateur

Ateliers UX/UI / UX/UI / Expérience Utilisateur

Complexité : 2 – Priorité : 2

L'objectif ici est de veiller à ce que l'interface utilisateur d'un service numérique ne recoure pas à des techniques manipulatoires, souvent désignées sous le terme de "dark patterns". Ces techniques sont des éléments de design qui cherchent à tromper ou à influencer les utilisateurs de manière insidieuse. Avec l'entrée en vigueur du Digital Services Act (DSA) en 2022, il est devenu impératif pour les fournisseurs de services numériques de concevoir des interfaces qui respectent l'autonomie des utilisateurs.

Moyens de mise en œuvre :

Les Designers Éthiques ont fourni un guide à ce sujet : <https://beta.designersethiques.org/fr/thematique-design-persuasif/concevoir-sans-dark-patterns>

Voici quelques exemples des pratiques à éviter :

- Créer des parcours de navigation complexes qui rendent difficile pour les utilisateurs d'effectuer certaines actions, comme se désabonner ou accéder à des informations importantes.
- Utiliser des publicités déguisées ou des pop-ups intrusifs qui apparaissent sans que l'utilisateur ait donné son accord.
- Présélectionner des options pour l'utilisateur, comme des cases à cocher, afin de l'inciter à accepter des conditions qu'il n'aurait pas choisies de son plein gré.
- Mettre en place des compteurs de temps pour des offres spéciales qui se réinitialisent, créant ainsi une fausse impression d'urgence. Pour aider à repérer ces éléments trompeurs, le guide de l'association Designers Éthiques peut être une ressource précieuse.

Critères de validation :

- L'interface utilisateur ne présente aucun dark pattern.
- Une évaluation régulière de l'interface est effectuée pour détecter et prévenir l'apparition de dark patterns.
- Les lignes directrices de la Commission européenne, notamment celles issues de l'article 25 du règlement 2022/2065, sont prises en compte dans cette évaluation.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=76

Ateliers UX/UI / UX/UI / Expérience Utilisateur

Complexité : 2 – Priorité : 3

Les « murs de contenu » et le défilement infini sont des techniques souvent utilisées pour capter l'attention des utilisateurs, mais elles ont des impacts négatifs sur l'expérience utilisateur et la bande passante. En effet, ces méthodes augmentent le temps passé sur la page, ce qui alourdit le poids de celle-ci et consomme plus de ressources. L'objectif est donc de réduire l'utilisation de ces techniques en favorisant des alternatives comme la pagination ou des boutons « Voir plus ».

Moyens de mise en œuvre :

- Évitez les « murs de contenus » et le défilement infini. Optez plutôt pour une navigation où le contenu se charge à la demande de l'utilisateur, par exemple avec un bouton « Voir plus » ou une pagination.
- Par défaut, affichez une liste de 10 éléments, mais laissez la possibilité à l'utilisateur d'en afficher 25, 50 ou 100 par page selon ses besoins.
- Remplacez le défilement infini par une action utilisateur, comme une pagination ou un bouton « Voir plus ». Assurez-vous de trouver le bon équilibre pour éviter de frustrer l'utilisateur avec trop de clics nécessaires.

Critères de validation :

- Le design du service numérique repose sur un chargement à la demande du contenu, proportionné au contexte d'utilisation (par exemple, un bouton « Voir plus » ou une pagination).
- Tous les tableaux et listes de résultats sont paginés.
- Le contenu peut s'afficher en entier sur un écran sans nécessiter de défilement infini.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=62
- Les 115 bonnes pratiques d'écoconception web : 13
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=52>

UX-UX-07 – Limiter et Personnaliser les Notifications

Ateliers UX/UI / UX/UI / Expérience Utilisateur

Complexité : 2 – Priorité : 2

Les notifications peuvent être une véritable frustration pour l'utilisateur si elles sont mal gérées. Il s'agit de trouver un équilibre entre tenir l'utilisateur informé et ne pas le solliciter en continu, en minimisant le nombre de notifications envoyées, en réfléchissant à leur pertinence et en donnant à l'utilisateur le contrôle total sur leur réception.

Moyens de mise en œuvre :

- **Pertinence** : Assurez-vous que chaque notification envoyée est réellement utile pour l'utilisateur. Évitez les messages redondants sur différents canaux (SMS, email, notification d'application, etc.).
- **Configuration par Défaut** : Proposez une configuration par défaut qui limite le nombre de notifications. Par exemple, visez un seuil inférieur à cinq notifications par jour.
- **Personnalisation** : Permettez à l'utilisateur de personnaliser les notifications. Ils devraient pouvoir choisir de qui, quand et comment ils reçoivent les notifications.
- **Formats et Urgence** : Réfléchissez aux différents formats de notifications en fonction de leur degré d'urgence. Par exemple, un SMS peut être plus approprié pour une information urgente qu'un courriel.
- **Confirmations** : Remplacez les emails de confirmation inutiles par des changements d'état visibles directement dans l'interface utilisateur. Par exemple, au lieu d'envoyer un courriel pour confirmer une désinscription, affichez simplement un message de confirmation dans l'application.
- **Changements d'État** : Remplacez les toasters de confirmation par des indicateurs visuels de changements d'état. Par exemple, au lieu d'afficher un toaster indiquant qu'un nouvel élément a été créé, mettez en évidence cet élément dans l'interface.

Critères de validation :

- Le service numérique ne propose pas de notifications ou propose par défaut un nombre de notifications limité (inférieur à cinq par jour).
- L'utilisateur a la possibilité de désactiver et de réduire, via son interface, les notifications proposées par le service (par exemple, un bouton directement visible sur l'interface utilisateur).
- Les possibilités de suppression ou de réduction des notifications sont mises en évidence.
- La fréquence et la quantité des notifications sont suivies.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=75
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=53>

UX-UX-08 – Informer l'utilisateur du format de saisie attendu lors de la soumission d'un formulaire

Ateliers UX/UI / UX/UI / Expérience Utilisateur

Complexité : 1 – Priorité : 1

L'objectif ici est de réduire le nombre d'échanges entre le client et le serveur en vérifiant les saisies directement sur le terminal de l'utilisateur. Ce faisant on diminue les risques d'erreurs lors de la soumission des formulaires et on évite des allers-retours inutiles vers le serveur pour corriger des erreurs de saisie.

Moyens de mise en œuvre :

Définir clairement les formats de saisie attendus et de les communiquer de manière explicite à l'utilisateur. Cela peut se faire par le biais de textes d'aide, d'infobulles, d'exemples de saisie ou de formats par défaut qui guident l'utilisateur dans sa saisie. En cas d'erreur, il est crucial de signaler visuellement quel champ est concerné et quel type de correction est nécessaire. La validation des saisies et des formats de données obligatoires doit être effectuée côté client, sans nécessiter de requête serveur, lorsque cela est possible.

Attention, cette validation côté frontend ne remplace pas la nécessité d'une validation côté backend, qui reste indispensable pour garantir l'intégrité des données.

Critères de validation :

- L'utilisateur reçoit des indications claires sur les formats de saisie attendus avant de soumettre le formulaire.
- Les messages d'erreur appropriés sont affichés lorsque l'utilisateur remplit le formulaire avec des données incorrectes ou manquantes.
- Les saisies et les formats de données obligatoires sont validés côté client avant la soumission du formulaire.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=72

UX-UX-09 – Informer l'utilisateur, avant le transfert, des poids et formats de fichier attendus dans les formulaires

Ateliers UX/UI / UX/UI / Expérience Utilisateur

Complexité : 2 – Priorité : 1

L'objectif ici est de réduire les échanges de fichiers lourds entre le client et le serveur en fournissant à l'utilisateur des informations claires sur les exigences relatives aux fichiers avant qu'il ne procède à leur transfert. Cela permet non seulement d'optimiser l'utilisation du réseau, mais aussi d'améliorer l'expérience utilisateur en évitant les frustrations liées à des fichiers non conformes.

Moyens de mise en œuvre :

- Pour les fichiers téléchargés, afficher clairement les informations concernant les poids et formats de fichier attendus, en particulier pour les fichiers dépassant 10 Mo. Cela inclut des détails tels que le format de fichier accepté et la taille maximale autorisée pour chaque type.
- S'assurer que l'utilisateur ne peut pas soumettre le formulaire tant que les prérequis de poids et de format ne sont pas respectés.
- Prendre en compte les formats de fichiers qui émergent, souvent plus efficaces en termes de compression.

Critères de validation :

- Les informations sur les poids et formats de fichier attendus sont correctement affichées à l'utilisateur avant le transfert.
- Le formulaire ne peut pas être soumis si les contraintes de poids et de format ne sont pas respectées.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=73

Ateliers UX/UI / UX/UI / Expérience Utilisateur

Complexité : 1 – Priorité : 3

L’objectif est de créer un environnement de lecture permettant à tous les utilisateurs d’accéder à l’information sans obstacles. L’utilisation de polices standards linéales, également connues sous le nom de polices sans serif, est fortement encouragée pour garantir une meilleure accessibilité des contenus numériques. Ces polices sont généralement plus faciles à lire, notamment pour les personnes rencontrant des difficultés visuelles ou d’autres handicaps. En effet, les polices cursives, manuscrites ou décoratives peuvent rendre la lecture laborieuse, ce qui peut nuire à l’expérience utilisateur. Il est également conseillé d’éviter les polices condensées, ainsi que l’utilisation excessive de majuscules, de soulignements ou d’italiques, qui peuvent compliquer la compréhension du texte. De plus, les éléments de texte animés ou défilants doivent être évités, car ils peuvent distraire ou désorienter les utilisateurs.

Moyens de mise en œuvre :

Choisir des polices adaptées dès le début du processus de conception. Cela implique de sélectionner des polices sans serif qui sont bien établies et largement reconnues pour leur lisibilité, comme Arial, Helvetica ou Verdana. Il est également important de définir une hiérarchie typographique claire, en utilisant différentes tailles et poids de police pour distinguer les titres, sous-titres et corps de texte.

Tester les choix typographiques sur différents appareils et tailles d’écran pour s’assurer qu’ils restent lisibles dans tous les contextes.

Critères de validation :

- Les polices utilisées sont exclusivement des polices sans serif.
- Aucune portion de texte n’est présentée en majuscules, en italique ou en souligné sur de longues sections.
- La hiérarchie typographique est clairement définie et respectée dans l’ensemble du contenu.
- Les textes animés ou défilants sont absents de l’interface.
- Des tests de lisibilité ont été réalisés avec des utilisateurs, incluant des personnes en situation de handicap, et les retours ont été pris en compte pour d’éventuels ajustements.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=44>

UX-UX-11 – Adapter le service à différents types d’affichages

Ateliers UX/UI / UX/UI / Performance et Optimisation

Complexité : 2 – Priorité : 3

L’idée ici est de concevoir un service numérique qui puisse fonctionner sur une variété de terminaux, y compris ceux avec des écrans plus petits, comme les smartphones anciens. En permettant à un service numérique de s’adapter à des écrans de différentes tailles et résolutions, on contribue à réduire la nécessité d’acheter de nouveaux appareils. Cela aide à lutter contre l’obsolescence des équipements causée par des logiciels qui ne sont pas optimisés pour les anciens terminaux.

Moyens de mise en œuvre :

Adopter l’approche du design réactif (« responsive design ») qui permet au service de s’afficher correctement sur des écrans de toutes tailles, que ce soit un petit smartphone ou un grand écran d’ordinateur. Plutôt que de créer des versions distinctes du service pour chaque type de terminal, il est préférable d’opter pour une solution unique qui s’adapte. Il faut également s’assurer que les menus et autres éléments d’interface soient utilisables aussi bien en mode tactile qu’avec un clavier. Dans certains cas, adopter une approche « mobile first » pour le développement peut favoriser une interface plus épurée et efficace.

Critères de validation :

- Les fonctionnalités essentielles du service numérique sont testées sur différents types d’affichage, notamment les ordinateurs de bureau, les tablettes et les mobiles.
- Le service doit démontrer une capacité d’adaptation dynamique de son affichage en fonction de la taille de l’écran, en respectant les principes du « responsive web design ».
- Il est nécessaire de vérifier que tous les composants, tels que les menus, sont accessibles via différents types d’interfaces, qu’il s’agisse d’un écran tactile ou d’un dispositif avec souris.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=42

UX-UX-12 – Utiliser majoritairement des composants fonctionnels natifs du système d'exploitation et du navigateur

Ateliers UX/UI / UX/UI / Performance et Optimisation

Complexité : 2 – Priorité : 1

L'objectif est de minimiser le recours à des composants développés en surcouche, qui peuvent alourdir le service et entraîner des temps de chargement plus longs, en favorisant l'utilisation de composants fonctionnels qui sont déjà intégrés dans le système d'exploitation, le navigateur ou le langage de programmation utilisé. Ces composants, comme les menus, les boutons ou les formulaires, sont conçus pour fonctionner de manière optimale avec les ressources disponibles. Cela peut également améliorer l'expérience utilisateur, car ces composants sont souvent plus familiers et intuitifs pour les utilisateurs.

Moyens de mise en œuvre :

Privilégier l'utilisation de composants natifs. En parallèle, il est recommandé de ne charger les ressources et les composants que lorsqu'ils sont réellement nécessaires, ce qui peut se faire par le biais de techniques comme le chargement à la demande ou le lazy loading. Cela permet non seulement d'optimiser la performance, mais aussi de réduire la sollicitation de ressources inutiles.

Critères de validation :

- Le service fait un usage majoritaire de composants fonctionnels natifs.
- En cas de recours à des composants non natifs, une évaluation de leur nécessité est effectuée, au regard des contraintes techniques, fonctionnelles et de l'impact de l'ajout de ces composants sur le système.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=67

UX-UX-13 – Favoriser un design simple, épuré, adapté au web

Ateliers UX/UI / UX/UI / Performance et Optimisation

Complexité : 3 – Priorité : 3

L'objectif est de privilégier un design simple et épuré, qui peut être réalisé uniquement avec HTML5 et CSS3 et avec le minimum d'ajout de module externe et de bibliothèques supplémentaires sur le Framework choisit, afin de garantir une expérience utilisateur fluide et agréable, sans surcharge de l'interface et sollicitation inutile de ressources matérielles. Penser au grand écart en termes de surcharge d'interface entre la page d'accueil de Google et Yahoo pour garder en tête la notion de "Gras fonctionnel" que nous cherchons à éviter à tout pris dans la démarche d'écoconception.

Moyens de mise en œuvre :

Se référer aux heuristiques d'ergonomie établies par des experts tels que Bastien et Scapin ou Nielsen, qui fournissent des principes directeurs pour concevoir des interfaces intuitives et efficaces. En parallèle, il faut tenir compte des contraintes techniques qui peuvent impacter la réalisation du design comme le choix d'un Framework UI comme Material ou Bootstrap.

Critères de validation :

- Les éléments visuels respectent les normes d'accessibilité et de lisibilité.
- Les retours des utilisateurs indiquent une satisfaction quant à la simplicité et l'efficacité du design.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 12

UX-UX-14 – Limiter les requêtes serveur lors de la saisie utilisateur en évitant l'autocomplétion

Ateliers UX/UI / UX/UI / Performance et Optimisation

Complexité : 2 – Priorité : 3

L'empreinte environnementale des services numériques est en partie due à la quantité de données échangées sur les réseaux. L'objectif ici est de réduire cette volumétrie de données en évitant les requêtes client/serveur inutiles, notamment lors de la saisie utilisateur dans des formulaires ou des suggestions de résultats. En limitant ces échanges, on peut souvent améliorer l'efficacité sans pour autant dégrader l'expérience utilisateur. L'autocomplétion, par exemple, envoie une requête au serveur à chaque caractère saisi, ce qui peut être coûteux en termes de ressources et inutile au regard de la vitesse de frappe au clavier. Préférez des méthodes comme la saisie assistée, qui guide l'utilisateur avec des informations locales, réduisant ainsi les échanges avec le serveur.

Moyens de mise en œuvre :

- Limitez la complétion automatique en ligne. Si vous devez absolument l'utiliser, attendez que l'utilisateur ait saisi au moins trois caractères et ajoutez un délai de 500 ms avant de lancer une requête réseau.
- Évitez les mécanismes d'autocomplétion ou de suggestions automatiques. Utilisez plutôt des techniques d'aide existantes comme la prévention des erreurs, des exemples ou des formats de saisie. Si l'autocomplétion est vraiment nécessaire, limitez-la en ne proposant des suggestions qu'après un certain nombre de caractères tapés.
- Utiliser des moteurs de recherche fulltext comme Meilisearch qui va gérer automatiquement les fautes de frappe, les synonymes et les traductions, et que vous pourrez configurer avec votre propre système de ranking. Ce genre d'outil permet de se concentrer sur la pertinence de ce qui est renvoyé à l'utilisateur en étendant et classifiant les résultats plutôt que de chercher à ce que l'utilisateur ne renseigne précisément ce qu'il recherche.

Critères de validation :

- Le service ne propose pas d'autocomplétion.
- Si l'autocomplétion est présente, elle doit être justifiée du point de vue de l'utilisateur et attend un délai d'au moins 500 ms avant de s'activer et nécessite au moins 3 caractères saisis.
- L'interface utilisateur permet de désactiver l'autocomplétion.
- Le nombre de requêtes HTTP entre client et serveur est contrôlé et suivi dans le temps.
- Il n'y a pas de requêtes identiques et redondantes.
- Les requêtes externes effectuées en appelant le site (en vidant le cache ou en utilisant une extension appropriée) sont effectivement nécessaires à l'exécution du service.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=71
- Les 115 bonnes pratiques d'écoconception web : 14
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=53>

UX-UX-15 – Limiter les animations javascript et CSS pour une meilleure performance et accessibilité

Ateliers UX/UI / UX/UI / Performance et Optimisation

Complexité : 2 – Priorité : 2

Les animations en Javascript et CSS, bien qu'attrayantes à première vue, peuvent être très gourmandes en ressources. Elles sollicitent fortement le CPU et la mémoire, déclenchant des actions coûteuses comme le (re)paint et le (re)flow. En plus de cela, elles peuvent poser des problèmes d'accessibilité et sont souvent désagréables pour l'expérience utilisateur à la longue. Nous conseillons de limiter les animations au strict nécessaire.

Moyens de mise en œuvre :

- Évitez les animations inutiles en Javascript. Par exemple, pour l'apparition d'un formulaire, faites-le apparaître directement sans effet de fondu.
- Si vous devez absolument utiliser des animations, privilégiez les propriétés CSS3 comme opacity et transform, ainsi que les fonctions translate, rotate et scale. Ces propriétés sont optimisées par les navigateurs et peuvent être prises en charge par le GPU, réduisant ainsi l'impact sur les performances.

```
```css
.box {
 will-change: transform, opacity;
}
```
```

- Pensez également à utiliser l'instruction will-change pour informer le navigateur qu'une animation va se produire, ce qui permet d'optimiser encore davantage les ressources. Enfin, respectez les préférences des utilisateurs en matière de mouvement réduit via la media query prefers-reduced-motion.
- Consultez des ressources comme <https://csstriggers.com> pour comprendre les actions sur le DOM déclenchées par une animation et <https://web.dev/animations-guide/> pour des conseils supplémentaires sur l'optimisation des animations.

Critères de validation :

- Le nombre d'animations JS / CSS par page est limité.
- Les animations utilisées se limitent aux propriétés CSS3 opacity et transform.
- Les préférences des utilisateurs en matière de mouvement réduit sont respectées via la media query prefers-reduced-motion.
- Les animations ne perturbent pas l'expérience utilisateur et n'ont pas d'impact négatif sur l'accessibilité.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 9
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=41>
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=43>

UX-UX-16 – Ne pas afficher les documents à l'intérieur des pages

Ateliers UX/UI / UX/UI / Performance et Optimisation

Complexité : 1 – Priorité : 3

L'intégration de documents directement dans les pages web peut sembler pratique pour certains rédacteurs de contenu, mais cela engendre souvent des complications inutiles. En effet, afficher des fichiers directement nécessite l'intégration de lecteurs spécifiques qui peuvent s'avérer lourds à mettre en place et surcharge l'affichage.

Pour l'utilisateur, le fait de charger un document à chaque consultation de la page peut être frustrant, la plupart préfèrent avoir la possibilité de télécharger le document uniquement s'ils en ont besoin.

Moyens de mise en œuvre :

Au lieu d'intégrer les documents à la page web, il est préférable d'insérer un lien de téléchargement qui permettra aux utilisateurs de récupérer le document s'ils le souhaitent, avec une description rapide du contenu de celui-ci et plusieurs options de téléchargement en fonction de la complexité de celui-ci. Se référer à la pratique UX-CO-09 "Fournir un résumé des documents disponibles au téléchargement".

Critères de validation :

- Aucun document n'est affiché directement dans l'interface.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 33

UX-UX-17 – Remplacer les cartes intégrées

Ateliers UX/UI / UX/UI / Performance et Optimisation

Complexité : 1 – Priorité : 2

L'utilisation de plugins comme Google Maps, qui pèse environ 1 Mo, soulève des interrogations quant à leur réelle valeur ajoutée sur un site. Tous les utilisateurs de mon site ont-ils réellement besoin de charger une carte interactive ? Surtout lorsque l'information qu'elle fournit pourrait être compensée par des alternatives moins lourdes. Parfois, une simple image de la carte accompagnée d'un lien vers une carte interactive peut suffire, surtout si l'on considère que tous les visiteurs n'utiliseront pas cette fonction.

Moyens de mise en œuvre :

- Réfléchir à l'utilité d'une carte pour les utilisateurs :
- Tous les utilisateurs ont-ils besoin de voir la carte ?
- La taille de la carte justifie-t-elle son poids en termes d'information apportée ?
- Peut-on envisager un lien vers une carte interactive que seuls les utilisateurs intéressés ouvriraient ?

Critères de validation :

- Les tests utilisateurs confirment que la carte est utile pour la navigation, ET
- Les données d'utilisation montrent que la majorité des visiteurs interagissent avec la carte ou le lien proposé, OU
- Une alternative (image ou lien) est en place et répond à leurs attentes sans nuire à leur expérience.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=46>

Ateliers UX/UI / UX/UI / Performance et Optimisation

Complexité : 2 – Priorité : 3

Les carrousels alourdissent les pages web en raison des CSS et JavaScript nécessaires, et chaque écran supplémentaire du carrousel y ajoute du poids. Ils compliquent également l'assurance qualité web et l'accessibilité numérique, et consomment plus de ressources processeur, surtout s'ils sont défilés automatiquement.

Les GIFs animés, quant à eux, sont des reliques de 1995 et sont souvent plus lourds et plus lents que des formats vidéo modernes comme WebM ou MP4.

En général, les animations doivent pouvoir être contrôlées par les utilisateurs pour une meilleure accessibilité et expérience utilisateur. Les carrousels automatiques et les GIFs animés non contrôlables sont donc à éviter.

Moyens de mise en œuvre :

- Préférer le contenu statique : Remplacez les carrousels par du contenu statique mis à jour régulièrement. Si vous devez absolument utiliser un carrousel, assurez-vous qu'il dispose de contrôles complets (arrêt, écran suivant/précédent).

- Remplacer les GIFs par des vidéos : Utilisez des formats vidéo modernes comme WebM ou MP4 à la place des GIFs animés. Voici un exemple de code pour intégrer une vidéo en autoplay, en boucle et muet :

```
```html
<!-- Autoplay, loop and muted video -->
<video autoplay loop muted playsinline>
 <source src="my-animation.webm" type="video/webm">
 <source src="my-animation.mp4" type="video/mp4">
</video>
```
```

Pour plus d'informations, consulter la page <https://web.dev/replace-gifs-with-videos/>

Critères de validation :

- Les carrousels sont inexistantes ou ne propose pas plus de 3 tuiles.
- Aucun GIF animés n'est utilisé.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 10
- Les 115 bonnes pratiques d'écoconception web : 98
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=41>

UX-UX-19 – Limiter le nombre de polices de caractères téléchargées

Ateliers UX/UI / UX/UI / Performance et Optimisation

Complexité : 1 – Priorité : 1

L'objectif est d'éviter de surcharger le service numérique avec des polices inutiles qui ralentissent le chargement et consomment des ressources. On privilégiera les polices standards déjà présentes sur l'ordinateur de l'utilisateur. S'il faut vraiment utiliser une police spécifique, il est plus intéressant de l'héberger soi-même, d'utiliser des techniques comme le faux-gras via la propriété CSS `-webkit-text-stroke`, et d'optimiser le fichier de police en l'allégeant des caractères non nécessaires.

Moyens de mise en œuvre :

Fixer comme objectif de n'utiliser au maximum que deux polices différentes et quatre variantes au total par page ou unité d'affichage.

Critères de validation :

- Le nombre de polices téléchargées est limité à deux, avec au maximum quatre variantes au total par page ou unité d'affichage.
- La taille des polices téléchargées ne dépasse pas 400 Ko par page ou unité d'affichage.
- On vérifie que les polices sont bien compressées et qu'on utilise uniquement les glyphes nécessaires.
- On s'assure que le mode de chargement des polices est optimisé pour ne pas bloquer le rendu de la page.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 32
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=70
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=44>
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=44>

UX-UX-20 – Limiter le nombre de champs de formulaires

Ateliers UX/UI / UX/UI / Performance et Optimisation

Complexité : 2 – Priorité : 2

L'objectif ici est de simplifier l'expérience utilisateur en ne demandant que les informations réellement indispensables, ce qui peut également contribuer à réduire le volume de données à envoyer, traiter et stocker.

Moyens de mise en œuvre :

Rendre un maximum de champs optionnels, voir ne pas les inclure du tout si la présence de cette information n'est pas strictement nécessaire au fonctionnement du service.

Par exemple, le prénom et le nom sont souvent obligatoires lors des inscriptions alors que ces données ne sont quasiment jamais utilisées par la suite.

Critères de validation :

- Le formulaire ne contient pas plus de trois champs obligatoires.
- Les champs facultatifs sont clairement identifiés et ne créent pas de confusion pour l'utilisateur.
- Aucune information jugée non essentielle n'est demandée.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=50>

UX-UX-21 – Éviter d'intégrer des fils d'actualité intégrés

Ateliers UX/UI / UX/UI / Réseaux sociaux

Complexité : 1 – Priorité : 3

Cette pratique peut alourdir le chargement de votre page et augmenter le nombre de trackers, ce qui sollicite davantage de ressources et réduit la vie privée de vos utilisateurs. Au lieu de cela, privilégiez l'option de créer des liens vers vos profils sur les réseaux sociaux ou sélectionnez avec soin les actualités que vous souhaitez mettre en avant. Cela permet d'améliorer l'expérience utilisateur en rendant votre site plus rapide et plus fluide.

Moyens de mise en œuvre :

- Analyser les besoins de votre audience pour déterminer quelles informations sont réellement pertinentes à afficher.
- Créer des liens directs vers vos réseaux sociaux, en veillant à ce qu'ils soient bien visibles et accessibles.
- Mettre en place un système de sélection manuelle des actualités à mettre en avant, en évitant les flux automatiques qui peuvent être encombrants.

Critères de validation :

- Le site n'intègre pas de fils d'actualité intégrés.
- Les liens vers vos réseaux sociaux sont cliqués et utilisés par les visiteurs, indiquant un intérêt pour ces plateformes.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=46>

UX-UX-22 – Préférer les alternatives aux chatbots

Ateliers UX/UI / UX/UI / Chatbot

Complexité : 1 – Priorité : 2

Les chatbots sont omniprésents sur les sites web, souvent présentés comme des solutions pratiques pour aider les utilisateurs, mais leur utilisation soulève plusieurs préoccupations. Ces outils, engendrent des appels à des modules et des serveurs externes et peuvent même recourir à des systèmes d'intelligence artificielle, ce qui alourdit considérablement le poids des pages et augmente le nombre de requêtes. Ils peuvent dégrader l'expérience de navigation, en créant des pop-ups intrusifs, des menus flottants qui masquent le contenu essentiel, ou des animations distrayantes qui perturbent la concentration de l'utilisateur. Ils peuvent générer de la frustration, notamment lorsqu'ils ne comprennent pas les demandes des utilisateurs ou lorsqu'ils sont inactifs en dehors des heures d'ouverture. Ils posent également des questions de confidentialité via le partage de données personnelles avec des entreprises tierces et l'acceptation de cookies. Et pour finir, le coût de mise en place et de maintenance de ces chatbots peut être relativement élevé. Il est souvent plus judicieux d'investir dans l'amélioration de l'expérience utilisateur sur le site, ce qui peut offrir un meilleur retour sur investissement. Trouver des alternatives aux chatbots est donc préférable dans la majorité des cas.

Moyens de mise en œuvre :

Commencer par une analyse approfondie des besoins des utilisateurs. Cela peut inclure des enquêtes, des interviews ou des tests d'usabilité pour comprendre les attentes et les frustrations des utilisateurs. Ensuite, explorer des alternatives aux chatbots, telles que des FAQ dynamiques, des guides interactifs, ou même des vidéos explicatives qui peuvent répondre aux questions courantes sans nécessiter d'interaction en temps réel. Il est également important de concevoir une interface utilisateur claire et intuitive, qui facilite la navigation et l'accès à l'information sans recourir à des éléments intrusifs.

Critères de validation :

- Le service ne propose pas de chatbot pour gérer les demandes utilisateurs.
- Les demandes d'assistance humaine diminuent, ce qui montre que les alternatives mises en place répondent efficacement aux besoins des utilisateurs.
- Les indicateurs de performance environnementale, tels que le poids de la page et le nombre de requêtes, montrent une amélioration par rapport à l'utilisation de chatbots.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=42>

Ateliers d'architecture logiciel, devops, hébergement

Ateliers au cours de laquelle l'équipe Scrum, en collaboration avec les architectes, explore et définit les grandes lignes techniques du produit, en s'assurant de leur alignement avec les besoins fonctionnels et les contraintes du projet.

AR-API-01 – Construire votre API avec des principes centrés sur le client

Ateliers d'architecture logiciel, devops, hébergement / API / Architecture d'API

Complexité : 3 – Priorité : 3

L'objectif est de concevoir une API qui facilite la vie des développeurs en leur fournissant les données dont ils ont besoin avec le moins d'appels possible. En effet, chaque requête envoyée à une API consomme des ressources, non seulement du côté du serveur qui doit la traiter, mais aussi du côté du réseau pour la transmettre.

Le but est de réduire l'empreinte environnementale de l'API en minimisant la quantité de données échangées et le nombre de requêtes nécessaires. Cela permet d'économiser des ressources réseau et de réduire la consommation énergétique des serveurs.

Moyens de mise en œuvre :

Structurer le contrat d'interface : Concevoir une structure de données qui regroupe logiquement les informations, afin de répondre aux besoins des clients qui vont consommer l'API en un minimum de requêtes. Si l'API donne des informations sur des concerts, par exemple, une seule requête devrait pouvoir renvoyer les détails du concert (date, lieu, artistes) ainsi que des informations sur la salle (capacité, accessibilité), plutôt que d'obliger à faire deux requêtes séparées.

Critères de validation :

- Les ateliers d'architectures ont intégré cette vision centrée sur le client dans la conception du contrat d'interface.
- La quantité d'appels aux API pour récupérer l'entièreté des informations pour afficher un même objet métier ne nécessite pas plus de 5 requêtes.

Liens :

- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=23

AR-API-02 – Privilégier JSON à XML pour les échanges de données

Ateliers d'architecture logiciel, devops, hébergement / API / Architecture d'API

Complexité : 1 – Priorité : 2

Pour minimiser l'empreinte d'une API, notamment du point de vue du tier réseaux, il est préférable de choisir un format d'échange de données léger.

Moyens de mise en œuvre :

- Évaluer les besoins spécifiques de l'API pour déterminer le format d'échange le plus adapté.
- Privilégier les formats comme JSON qui offrent une meilleure efficacité en termes de taille des données.
- Adapter les structures de données pour optimiser l'utilisation du format choisi.

Critères de validation :

- Le format d'échange choisi est plus léger que les alternatives.

Liens :

- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=19

AR-API-02 – Minimiser le taux d'erreur des API

Ateliers d'architecture logiciel, devops, hébergement / API / Erreurs HTTP

Complexité : 3 – Priorité : 3

En réduisant le taux d'erreur (les requêtes qui ne renvoient pas un code HTTP 2xx), c'est-à-dire en diminuant le nombre de réponses qui ne sont pas des succès, on évite un traitement excessif des requêtes.

Moyens de mise en œuvre :

Commencez par analyser les logs de votre application afin d'identifier les types d'erreurs les plus fréquentes. Vous pouvez par exemple mettre en place des validations côté client (vérifications de format, contrôles de champs obligatoire) pour vous assurer que les données envoyées sont correctes avant même qu'elles n'atteignent le serveur.

Critères de validation :

- Le taux d'erreur global est mesuré et se situe en dessous d'un seuil prédéfini.
- Des validations côté client sont en place et fonctionnent correctement, avec des messages d'erreur clairs pour les utilisateurs.
- Des tests réguliers des API sont effectués.

Liens :

- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=33

AR-AR-01 – Choisir entre site statique et CMS

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Conception et de Choix Technologiques
Complexité : 3 – Priorité : 3

Le choix entre un site statique et un CMS (Content Management System, ou Système de Gestion de Contenu) est souvent un casse-tête.

Les sites statiques génèrent chaque page HTML individuellement, séparant la forme et le contenu conformément aux recommandations du W3C. Cela permet une meilleure gestion éditoriale, améliore les performances et réduit l'impact environnemental du service. Cependant, il peut y avoir des défis liés à la maintenance des technologies utilisées par les générateurs de sites statiques.

Les CMS, quant à eux, permettent de créer et de gérer un site web facilement, même sans compétences techniques. Ils offrent des fonctionnalités comme la gestion des droits des administrateurs et la structuration du contenu. Cependant, ils sont souvent gourmands en ressources et surdimensionnés pour certains besoins.

Moyens de mise en œuvre :

Les questions à se poser :

- Qui va créer et maintenir le service ?
- À quelle fréquence le contenu doit-il être mis à jour ?
- Quelles parties du contenu doivent être mises à jour ?
- Combien de temps le service va-t-il durer ? A-t-il déjà une fin de vie prévue ?

Site statique :

- Idéal pour les sites avec peu ou pas de mises à jour de contenu.
- Complexe pour l'édition de contenu, surtout pour les non-initiés.
- Utiliser un générateur de site statique adapté à vos besoins.

CMS :

- Pertinent si le contenu doit être mis à jour régulièrement par des personnes peu initiées.
- Choisir entre un éditeur WYSIWYG (What You See Is What You Get) pour plus de liberté graphique mais potentiellement moins d'écoconception, ou l'édition via un système textuel comme le markdown pour plus de sobriété.

Critères de validation :

- Le site repose sur un CMS qui n'utilise pas plus de 3 extensions.
- Le site utilise un thème léger ou écoconçu.
- Les contributeurs sont formés à l'optimisation des images et aux bonnes pratiques d'écoconception.
- Un cache HTTP (reverse proxy) est mis en place pour les pages dynamiques d'un CMS.
- Les constructeurs de pages sont évités pour minimiser l'impact et améliorer les performances.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=25>
- Les 115 bonnes pratiques d'écoconception web : 22

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Conception et de Choix Technologiques
Complexité : 3 – Priorité : 3

Pour un service numérique, opter pour une PWA plutôt qu'une application mobile native peut être une solution plus écoresponsable. Les PWA offrent une expérience utilisateur fluide tout en minimisant les ressources nécessaires pour le développement et la maintenance. Elles permettent également d'éviter l'obsolescence rapide des équipements, car elles sont compatibles avec la plupart des navigateurs, même sur des appareils plus anciens.

Moyens de mise en œuvre :

- Analyser les besoins des utilisateurs pour déterminer si une PWA est adaptée.
- Développer une PWA responsive et performante en utilisant des technologies web modernes.
- Assurer une intégration fluide avec les fonctionnalités du navigateur et des appareils mobiles.

Critères de validation :

- La PWA couvre les mêmes fonctionnalités principales qu'une éventuelle application native.
- Les utilisateurs peuvent recevoir des notifications et utiliser certaines fonctionnalités clés hors ligne.
- Les performances de la PWA sont comparables à celles d'une application native.
- La PWA est accessible depuis différents navigateurs et appareils, y compris les plus anciens.
- L'utilisation de la PWA ne nécessite pas de matériel récent ou des mises à jour fréquentes du système d'exploitation. La notion de matériel récent est à définir dans la BP ECO-SP-01 "Assurer la compatibilité avec les anciens appareils et logiciels".

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=20>
- Les 115 bonnes pratiques d'écoconception web : 19

AR-AR-03 – Utiliser la version la plus récente du langage et de la plateforme

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Conception et de Choix Technologiques
Complexité : 3 – Priorité : 3

Les langages de programmation côté serveur comme PHP, Ruby, Java, et bien d'autres, évoluent constamment grâce aux efforts des communautés de développeurs. Chaque nouvelle version apporte son lot d'améliorations : meilleures performances, gestion de mémoire optimisée, stabilité accrue, et correction des failles de sécurité. En gros, c'est comme passer d'une vieille voiture à un modèle flambant neuf : tout fonctionne mieux et plus rapidement. Il en va de même pour les plateformes comme Node.js ou les Framework comme Spring en Java. Les mises à jour régulières apportent des améliorations significatives. Donc, pour tirer le meilleur parti de votre code et de vos services numériques, il est fortement recommandé d'utiliser la version la plus récente du langage et de la plateforme.

Moyens de mise en œuvre :

Par exemple, les versions 7 et 8 de PHP ont apporté des gains de performance considérables. PHP 8.0, avec sa fonctionnalité JIT (Just in Time Compilation), peut compiler des parties du code et les stocker en mémoire, ce qui améliore les performances et optimise la gestion de la mémoire. Suivez les mises à jour officielles des langages et des plateformes que vous utilisez, et planifiez des mises à jour régulières de votre environnement de développement et de production.

Critères de validation :

- Le service n'utilise pas de langage ou Framework avec plus de deux versions majeures de retard sur la dernière version stable.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 29

AR-AR-04 – Choisir les technologies les plus adaptées, standards et interopérables

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Conception et de Choix Technologiques
Complexité : 3 – Priorité : 3

Un choix judicieux des technologies est essentiel pour garantir la durabilité de votre service numérique. Il s'agit de sélectionner les outils les plus économes en fonction de vos besoins et contraintes métier. En optant pour des technologies standard et interopérables, vous luttez contre l'obsolescence des équipements induite par le logiciel. Cela permet d'allonger la durée de vie des terminaux. Par exemple, privilégier un site statique pour un contenu peu changeant ou une SPA (Single Page Application) pour des services dynamiques peut grandement optimiser les performances et l'allocation de ressources.

Moyens de mise en œuvre :

- Assurez-vous que les API/Framework utilisées sont standard et supportées par les navigateurs ou les systèmes qui devront intégrer le service.
- Si une application native est nécessaire, par exemple pour des traitements de données sensibles, assurez-vous qu'elle utilise des standards compatibles avec les principaux systèmes d'exploitation.
- Utilisez des composants open source au maximum pour construire votre service. Cela vous permet de garder la main sur la maintenance du code et d'améliorer sa durabilité.

Nos conseils pour le choix de technologies opensource : s'assurer de la popularité du repository de code utilisé, s'assurer que l'auteur permet de manière durable de contribuer au code, vérifier si la communauté est active (nombre de PR, d'issues, fréquence de mise à jour, nombre de fork).

Critères de validation :

- Le service s'appuie sur des standards interopérables communs aux principaux écosystèmes (terminaux, systèmes d'exploitation, navigateurs, etc.).

Liens :

- Les 115 bonnes pratiques d'écoconception web : 66
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=33

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Conception et de Choix Technologiques
Complexité : 3 – Priorité : 3

Le choix des protocoles réseau est important car il impacte les infrastructures et joue un rôle clé dans la durée de vie du service. Si les protocoles utilisés deviennent obsolètes, le service devient inutilisable. Pour éviter cela, il faut absolument choisir des protocoles qui minimisent les besoins de mises à jour et de modernisations. Par exemple, l'adoption de l'IPv6 est une bonne pratique, car l'IPv4 est en voie de disparition. De même, HTTPS est désormais indispensable pour la sécurité et l'accès aux services web. Enfin, HTTP/2 offre des avantages significatifs en termes de performance et d'efficacité par rapport à HTTP/1.1, grâce à son utilisation de la compression des en-têtes et du multiplexage des échanges.

Moyens de mise en œuvre :

Choix des protocoles :

- Protocole d'adressage : Assurez-vous que votre service est accessible en IPv6.
- Protocole d'authentification et de chiffrement : Utilisez HTTPS au lieu de HTTP. C'est souvent obligatoire, surtout pour le transfert de données personnelles (conformément à l'article 32 du RGPD). Assurez-vous que la version de TLS utilisée est la plus récente (par exemple, TLS v1.3).
- Privilégier HTTP/2 : HTTP/2 utilise une représentation binaire des requêtes et réponses, avec un mécanisme de compression des en-têtes (HPACK). Grâce au multiplexage, il permet de n'utiliser qu'une seule connexion TCP avec le serveur, ce qui optimise les performances. Si un autre nom de domaine redirige vers la même adresse IP et est compatible avec le certificat TLS, la connexion TCP peut être réutilisée, ce qui permet d'utiliser un nom de domaine distinct sans cookie pour les ressources statiques.

Critères de validation :

- Les différents composants du service numérique fonctionnent bien en IPv6 et ne font appel à aucun service tiers IPv4-only.
- Le service utilise HTTPS et non HTTP.
- La dernière version de TLS (au moment de la rédaction de ce référentiel : TLS v1.3) est bien supportée.
- Aucune requête utilisant HTTP/1.0 n'est effectuée.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=53
- Les 115 bonnes pratiques d'écoconception web : 82

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Conception et de Choix Technologiques
Complexité : 3 – Priorité : 3

Les plateformes cloud se sont imposées comme des acteurs incontournables grâce à leur capacité à offrir des infrastructures qui sont résilientes, évolutives, efficaces, sécurisées et mutualisées. Ces plateformes mettent à disposition une variété de services managés qui tirent parti de ces infrastructures. Ce sont des solutions parfaites pour le stockage de contenus Big Data, la persistance des données via des bases de données managées, la gestion de services de stockage et de flux de messages, comme Kafka, et la gestion de conteneurs, avec des outils comme Kubernetes.

Pour les équipes de développement, il est souvent judicieux de recourir à ces services managés en considérant qu'il est généralement difficile d'atteindre un niveau d'efficacité comparable à celui de ces solutions, surtout si l'on considère le temps et les ressources nécessaires pour développer et maintenir une infrastructure équivalente.

Moyens de mise en œuvre :

Les hébergeurs modernes proposent des solutions de « serverless computing », un modèle où les ressources nécessaires au fonctionnement d'un service sont allouées de manière dynamique. Cela signifie que les équipes responsables du service peuvent se concentrer davantage sur le développement et l'amélioration de leurs applications, sans avoir à se soucier des détails techniques de l'infrastructure sous-jacente. En parallèle, les hébergeurs optimisent l'affectation des ressources, ce qui permet de réduire les coûts et d'améliorer l'efficacité énergétique. Ce modèle est particulièrement avantageux pour les entreprises qui cherchent à innover rapidement tout en minimisant l'allocation de ressources inutiles.

Critères de validation :

- Les équipes de développement peuvent démontrer une compréhension claire des avantages des services managés.
- Les performances des applications utilisant des services managés sont mesurées et jugées satisfaisantes par rapport aux objectifs fixés.
- Une évaluation des coûts montre que l'utilisation de services managés est plus économique que le développement d'une solution interne équivalente.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 96

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Conception et de Choix Technologiques
Complexité : 2 – Priorité : 3

L'idée ici est de bien réfléchir à la qualité de service (QoS) et au niveau de disponibilité (SLA) que l'on souhaite offrir aux utilisateurs d'un site web ou d'un service en ligne. Par exemple, il n'est pas nécessaire d'investir dans un hébergement ultra-performant, comme un centre de données de type Tier IV, si les utilisateurs sont tout à fait satisfaits avec un taux de disponibilité de 99 % pour un service qui n'est pas critique. Plus un centre de données est conçu pour être disponible en permanence, plus son impact environnemental est important, car cela implique des systèmes redondants et actifs en permanence. L'objectif est d'éviter la sur qualité, c'est-à-dire de ne pas offrir plus que ce qui est réellement nécessaire et attendu. Il est préférable d'adapter la qualité de service en fonction des ressources disponibles et du niveau de disponibilité qui est réellement requis.

Moyens de mise en œuvre :

Pour mettre cela en pratique, on peut s'inspirer des grandes entreprises du web qui, malgré leur immense base d'utilisateurs, ne choisissent pas toujours des niveaux de disponibilité extrêmes comme ceux offerts par les centres de données de type Tier IV. En général, ces entreprises optent pour une redondance des données sur au moins un autre serveur situé dans un autre centre de données. Ainsi, si un serveur rencontre un problème, les utilisateurs sont automatiquement redirigés vers un serveur de secours. Ce processus de basculement peut prendre quelques secondes, ce qui est généralement acceptable et souvent imperceptible pour l'utilisateur final.

Critères de validation :

- Les analyses de performance montrent que le service fonctionne de manière fluide et efficace, sans interruptions notables, tout en respectant les engagements de disponibilité convenus.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 90

AR-AR-08 – Concevoir une architecture utilisant des composants eux-mêmes écoconçus

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Conception et de Choix Technologiques
Complexité : 3 – Priorité : 3

Nous souhaitons nous assurer que le service numérique que nous développons ne repose pas uniquement sur des choix techniques intelligents, mais aussi sur une réflexion globale concernant l'impact de toute les architectures nécessaires au service et des composants utilisés. Nos services s'appuient sur des éléments qui ne sont pas toujours sous notre contrôle, comme des Framework ou des bibliothèques développés par d'autres équipes ou d'autres structures. Il est donc tout aussi important dans notre démarche de vérifier que ces dépendances sont également conçues dans une optique de réduction de la sollicitation des ressources et de l'empreinte.

Moyens de mise en œuvre :

- Examinez les Framework utilisés pour le développement, tant pour le frontend que pour le backend. Les bonnes questions à se poser sont : sont-ils optimisés pour utiliser les ressources de manière efficace ? Utilisent-ils des techniques de compression qui réduisent la taille des données échangées ? Sont-ils capables d'optimiser les requêtes entre le client et le serveur pour minimiser la consommation d'énergie ?
- Vérifiez les composants et bibliothèques internes et externes que vous implémentez dans nos services. Assurez-vous qu'ils respectent nos principes d'écoconception. Cela implique d'analyser leur efficacité en matière de ressources matérielles et énergétiques.

Critères de validation :

- L'analyse de l'impact des composants et des Framework a été réalisée, et des configurations d'optimisations ont été utilisés lors de leur implémentation.
- Il a été vérifié que le service ne repose pas sur des technologies particulièrement énergivores, comme l'apprentissage automatique, la blockchain ou les applications de métavers.
- En cas d'utilisation de technologies non frugales, une documentation claire explique le choix effectué, en mettant en avant la solution la moins consommatrice de ressources.
- La déclaration d'écoconception du service numérique inclut une analyse des choix d'architecture et de composants.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=50

AR-AR-09 – Créer une architecture applicative modulaire

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Construction de l'Architecture
Complexité : 3 – Priorité : 3

L'idée derrière une architecture modulaire, très prisée dans le monde des logiciels open source, c'est de rendre les applications plus flexibles et plus faciles à maintenir. Si votre site web ou service en ligne est amené à évoluer, il vaut mieux construire un noyau solide avec les fonctionnalités essentielles en se basant sur votre définition du MVP, puis ajouter des modules indépendants les uns des autres au fur et à mesure. Ces modules peuvent regrouper des fonctions spécifiques à un domaine métier, ce qui permet de les développer indépendamment des autres parties de l'application. Si votre service est vendu par options à vos clients, cela permet ultimement de fournir le service numérique le plus léger possible pour répondre aux besoins métiers spécifiques de vos utilisateurs.

Moyens de mise en œuvre :

Nous pouvons nous inspirer des logiciels open source comme Nginx, Apache ou même l'ERP Odoo, qui utilisent tous cette approche.

Côté backend, le découpage en microservices est la meilleure manière d'apporter un haut niveau de modularité aux services HTTP. Attention toutefois à ne pas trop fragmenter les services, car cela pourrait ultimement augmenter les ressources informatiques nécessaires. L'idée, c'est de trouver le bon équilibre pour que chaque module soit suffisamment autonome sans que l'ensemble devienne ingérable. Redit a transité d'une architecture monolithique vers des microservices et est un bon exemple pour s'inspirer : <https://blog.bytebytego.com/p/reddits-architecture-the-evolutionary>

Critères de validation :

- Chaque fonctionnalité de base est implémentée dans un noyau central.
- Les modules complémentaires sont développés indépendamment et peuvent être partagés avec d'autres applications.
- La granularité du découpage en microservices est optimisée pour éviter une complexité excessive et une surconsommation de ressources.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 28

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Construction de l'Architecture
Complexité : 2 – Priorité : 2

Lorsqu'on utilise un CMS (système de gestion de contenu), le choix du thème est crucial car il s'accompagne souvent de l'ajout d'extensions et parfois même d'un constructeur de pages. Ces fonctionnalités supplémentaires dépassent souvent les besoins réels du site. Chaque extension ajoute des données dans la base, des fichiers CSS et JavaScript, et augmente le nombre de fichiers stockés sur le serveur. Selon la qualité du développement, ces éléments supplémentaires peuvent être lourds (voir même redondants si les mêmes librairies javascript sont utilisées), ce qui entraîne une augmentation du nombre de requêtes, du poids des pages, du temps de traitement des requêtes et une complexification de l'administration des contenus.

Moyens de mise en œuvre :

Pour éviter les problèmes liés à l'utilisation excessive d'extensions et de fonctionnalités inutiles, il est recommandé de choisir un thème simple qui contient uniquement l'essentiel. Un thème tout-en-un ou intégrant un constructeur de pages peut ajouter des fonctionnalités comme des carrousels, des boutons d'action ultra personnalisés, des témoignages ou des portfolios, même si vous n'en avez pas besoin. Il est également important de faire attention aux incompatibilités entre certaines extensions. La meilleure approche est de commencer avec un thème basique et d'ajouter des fonctionnalités supplémentaires uniquement après une réflexion approfondie sur leur nécessité.

Critères de validation :

- Les fonctionnalités ajoutées correspondent réellement aux besoins du site.
- Les extensions installées sont compatibles entre elles et ne se chevauchent pas en termes de fonctionnalités.
- Le temps de chargement des pages est mesuré et optimisé.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 68

AR-AR-11 – S'assurer qu'une seule API répond à un besoin spécifique

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Construction de l'Architecture
Complexité : 2 – Priorité : 3

Dans un système d'information, particulièrement dans les grandes infrastructures, il est fréquent d'avoir à disposition des API qui semblent avoir le même but et les mêmes objectifs. Ces API en double peuvent non seulement semer la confusion chez les utilisateurs, mais elles entraînent également une consommation de ressources superflues. Si une API en place couvre partiellement le champ fonctionnel que vous visez, il peut être judicieux de prendre contact avec son producteur pour l'adapter ou la compléter.

Moyens de mise en œuvre :

- Identifier le besoin spécifique que vous souhaitez adresser avec votre API.
- Consultez le catalogue de données de votre organisation pour rechercher des API existantes qui pourraient répondre à ce besoin.
- Analysez les fonctionnalités des API trouvées pour déterminer si l'une d'elles peut être adaptée ou étendue pour couvrir votre besoin.
- Si une API existante est pertinente, engagez une discussion avec son équipe de développement pour explorer les possibilités d'évolution.
- Si une unique fonctionnalité pertinente existe au sein d'une API qui n'adresse pas votre besoin métier (par exemple "envoyer un mail"), il peut être pertinent d'extraire cette fonctionnalité dans un microservices qui sera à l'avenir utilisé par toutes les API en ayant le besoin.
- Si aucune API ne correspond, documentez clairement le besoin et les raisons pour lesquelles une nouvelle API est nécessaire avant de procéder à son développement.

Critères de validation :

- Une recherche dans le catalogue de données a été effectuée.
- Une communication a eu lieu avec les producteurs d'API existantes pour discuter des possibilités d'évolution le cas échéant.
- La décision de créer une nouvelle API a été justifiée par des critères clairs et documentés.

Liens :

- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=16

AR-AR-12 – Utiliser OData ou GraphQL pour construire les API de vos bases de données

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Construction de l'Architecture
Complexité : 3 – Priorité : 2

Dans le monde du développement d'API, il est assez fréquent de voir des systèmes de backend conçus pour faciliter la consommation de bases de données. Cependant, il arrive souvent que ces systèmes soient complètement redéveloppés avec des schémas de données qui ne correspondent pas vraiment aux besoins des utilisateurs. Cela peut entraîner des situations où les utilisateurs doivent effectuer plusieurs requêtes, souvent complexes, pour obtenir toutes les informations dont ils ont besoin. Pour éviter ces désagréments et optimiser l'interaction avec les bases de données, il est fortement recommandé d'utiliser des technologies comme OData ou GraphQL. Ces outils permettent aux consommateurs de formuler des requêtes complexes de manière plus intuitive et efficace. En adoptant ces technologies, on s'attend à réduire l'impact sur le réseau, le calcul et le stockage.

Moyens de mise en œuvre :

Décidez si OData ou GraphQL est le mieux adapté à votre projet. OData est souvent plus simple à mettre en place pour des requêtes standardisées, tandis que GraphQL offre une flexibilité accrue pour des requêtes plus complexes.

Critères de validation :

- Les utilisateurs peuvent effectuer des requêtes complexes sans avoir à multiplier les appels à l'API.
- L'API répond rapidement aux requêtes, avec des temps de réponse optimisés.
- Les schémas de données sont bien documentés et compréhensibles pour les utilisateurs.
- Des tests de performance montrent une réduction significative de l'impact sur le réseau, le calcul et le stockage par rapport à une API traditionnelle.

Liens :

- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=25

AR-AR-13 – Utiliser un cache HTTP

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Construction de l'Architecture
Complexité : 3 – Priorité : 3

L'idée ici est de tirer parti des reverse proxies, comme Varnish, Squid ou Nginx, qui sont des outils vraiment efficaces pour servir du contenu web, que ce soient des pages HTML, des images ou d'autres types de fichiers. Ces solutions sont conçues pour être rapides tout en minimisant l'utilisation des ressources processeur. En évitant de surcharger le serveur d'applications avec des requêtes répétées pour le même contenu, on peut réduire la taille de l'infrastructure matérielle nécessaire pour délivrer le même service.

Moyens de mise en œuvre :

Optez pour un reverse proxy spécialisé comme Varnish, Squid ou Nginx.

Critères de validation :

- Tous les serveurs délivrant des ressources ou des données ont mis en place un cache HTTP.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 73

AR-AR-14 – Utiliser certains forks applicatifs orientés "performance"

Ateliers d'architecture logiciel, devops, hébergement / Architecture / Optimisation

Complexité : 3 – Priorité : 3

Les logiciels open source sont souvent dérivés, ou "forkés", pour améliorer leurs performances. Ces versions optimisées peuvent offrir des gains significatifs en termes de rapidité et d'efficacité, ce qui se traduit généralement par une réduction de la consommation de ressources. Si un fork optimisé existe et qu'il répond aux besoins fonctionnels et techniques de votre projet, il est recommandé de l'adopter pour bénéficier de ces améliorations.

Moyens de mise en œuvre :

- Pour les projets utilisant Drupal, il est conseillé de se tourner vers Pressflow, une version optimisée de Drupal qui offre de meilleures performances.
- Pour les projets utilisant Redis, il est recommandé d'opter pour KeyDB, une version optimisée de Redis qui propose des améliorations notables en termes de performance.

Critères de validation :

- Le projet utilise des forks orientés performance quand ceux-ci existent.
- Le fork utilisé doit offrir un périmètre fonctionnel et technique suffisant pour répondre aux besoins du projet.
- Une évaluation comparative des performances entre la version standard et la version forkée doit démontrer un gain significatif en termes de consommation de ressources.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 67

AR-BE-01 – Mettre les caches entièrement en RAM

Ateliers d'architecture logiciel, devops, hébergement / Backend / Optimisation des performances
Complexité : 2 – Priorité : 2

L'objectif est de maximiser l'utilisation de la mémoire vive (RAM) pour les systèmes de cache. En plaçant les caches en RAM, on évite les lenteurs liées aux entrées/sorties sur des supports de stockage persistants, plus lents en lecture/écriture comme les disques durs. Cela permet non seulement de répondre plus rapidement aux requêtes des utilisateurs, mais aussi de réduire le nombre de composants matériels et logiciels nécessaires pour traiter ces requêtes. La RAM, extrêmement rapide en termes de lecture et d'écriture, permet de minimiser le temps de consommation des ressources. Cette approche contribue à prolonger la durée de vie des composants, car elle élimine les mouvements mécaniques associés aux disques durs.

Moyens de mise en œuvre :

Pour intégrer un cache RAM dans un système comme Drupal, vous pouvez utiliser Memcache, un système de cache distribué (<https://drupal.org/project/memcache>). Une autre option est d'utiliser Varnish, un accélérateur HTTP qui peut également être intégré à Drupal (<https://drupal.org/project/varnish>).

Critères de validation :

- Tous les systèmes de cache en place sont effectivement configurés pour utiliser la RAM.
- Les performances du système montrent une amélioration mesurable en termes de temps de réponse et de charge serveur.
- Aucune opération d'entrée/sortie sur des disques durs n'est nécessaire pour les requêtes traitées par le cache.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 72

AR-BE-02 – Utiliser un serveur asynchrone

Ateliers d'architecture logiciel, devops, hébergement / Backend / Optimisation des performances
Complexité : 3 – Priorité : 3

L'utilisation de serveurs asynchrones, comme Nginx, Node.js ou Gwan, représente une approche intelligente pour optimiser la consommation des ressources. Contrairement aux serveurs traditionnels qui créent un nouveau processus ou un thread pour chaque requête, les serveurs asynchrones fonctionnent de manière à traiter plusieurs requêtes simultanément sans avoir à multiplier les ressources. Cela signifie qu'ils consomment moins de mémoire vive et restent stables même sous une forte charge.

Moyens de mise en œuvre :

Il est recommandé de choisir un serveur asynchrone reconnu pour sa performance, comme Nginx. Ce dernier est souvent cité comme étant plus efficace qu'Apache, car il peut traiter jusqu'à 2,1 fois plus de requêtes par seconde.

Critères de validation :

- Les performances du serveur asynchrone sont mesurées et montrent une capacité à traiter un nombre élevé de requêtes par seconde.
- La consommation de mémoire vive est surveillée et reste stable même lors de pics de trafic.
- Les temps de réponse des requêtes sont optimisés et conformes aux attentes de performance.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 85

AR-BE-03 – Réduire au nécessaire les logs des serveurs et collecter uniquement les données requises

Ateliers d'architecture logiciel, devops, hébergement / Backend / Optimisation des performances

Complexité : 2 – Priorité : 2

Les logs des serveurs, qu'ils soient web, applicatifs ou de base de données, peuvent rapidement devenir volumineux. Cela peut entraîner une consommation excessive de cycles CPU et un espace de stockage important. En configurant précisément le niveau de logs de l'application (par exemple, en ne traçant que les informations pertinentes comme les messages d'information, d'avertissement et d'erreur), on peut éviter des écritures massives sur le disque.

Moyens de mise en œuvre :

- Définir une politique de rétention des logs en fonction des besoins d'exploitation du service. Par exemple, conserver les logs pendant 30 jours puis les archiver ou les supprimer.
- Mettre en place des scripts ou des outils de nettoyage automatique des données pour supprimer régulièrement les informations obsolètes ou inutiles.
- Utiliser des solutions de gestion des logs comme ELK (Elasticsearch, Logstash, Kibana) pour centraliser, analyser et gérer les logs de manière efficace.

Critères de validation :

- Une politique de rétention des logs est définie et mise en œuvre.
- Des scripts ou des outils de nettoyage automatique des données de log sont en place et fonctionnent correctement.
- Le volume de données stockées est dans les logs est monitoré.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 86
- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=34

AR-BE-04 – Mettre en place des "Circuit breaker" sur les services tiers et les microservices non critiques

Ateliers d'architecture logiciel, devops, hébergement / Backend / Fiabilité et résilience

Complexité : 2 – Priorité : 2

L'idée derrière un circuit breaker est de créer un mécanisme intelligent qui permet de gérer les appels à un service tier quand ce dernier rencontre des problèmes. Imaginez que vous essayez d'accéder à un service en ligne, mais qu'il est en panne ou qu'il met trop de temps à répondre. Au lieu de continuer à le solliciter sans relâche, ce qui pourrait aggraver la situation, le circuit breaker intervient. Il évalue plusieurs critères, comme le temps d'attente ou le nombre d'erreurs rencontrées, et décide de ne plus faire appel à ce service pendant un certain temps. À la place, il peut renvoyer une valeur par défaut ou rediriger vers un autre service qui fonctionne. Cela permet non seulement de préserver les ressources, en évitant des appels réseau inutiles, mais aussi de donner au service défaillant le temps de se rétablir sans être submergé de requêtes.

Moyens de mise en œuvre :

Prenons l'exemple d'une page web essayant d'afficher une carte délivrée par un service tiers. Si ce service rencontre un problème et ne répond pas correctement, le service principal qui gère la page peut "apprendre" que ce service est en panne. Si cette carte n'est pas critique pour la réalisation de l'acte de l'utilisateur, plutôt que de continuer à l'appeler à chaque fois qu'un utilisateur consulte la page, le service va éviter de le solliciter tant qu'il ne fonctionne pas à nouveau. Même si certaines informations ne sont pas disponibles, la fonctionnalité critique reste opérationnelle et l'utilisateur peut toujours naviguer, bien que l'expérience soit un peu moins riche.

Critères de validation :

- Les services tiers et les microservices non critiques disposent de "circuit breaker".
- Les temps de réponse des services sont mesurés et montrent une réduction des appels en cas d'erreur.
- Les utilisateurs ne rencontrent pas de dégradations significatives de l'expérience lors de pannes de services.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 26

AR-BE-05 – Utiliser tous les niveaux de cache du CMS

Ateliers d'architecture logiciel, devops, hébergement / Backend / Optimisation CMS

Complexité : 3 – Priorité : 3

L'idée est de tirer parti des différentes couches de cache que propose un système de gestion de contenu (CMS) comme WordPress, Drupal ou eZ Publish. En utilisant ces niveaux de cache, on peut vraiment optimiser les performances de notre site tout en réduisant la consommation de ressources, que ce soit en termes de cycles CPU ou de mémoire.

Moyens de mise en œuvre :

Un CMS comme WordPress ne propose pas de système de cache intégré par défaut. Cependant, il existe une multitude d'extensions qui peuvent combler cette lacune. Parmi les plus populaires, on trouve W3 Total Cache, WP Super Cache et WP Rocket. En installant l'une de ces extensions, vous allez pouvoir mettre en cache les pages sur le serveur, ce qui signifie qu'elles ne seront pas générées à chaque fois qu'un utilisateur les demande, le cache coté data, qui ne seront pas requêtes à chaque demande, le cache coté front, qui ne cherchera pas à récupérer les ressources à chaque chargement. Ces outils offrent souvent d'autres optimisations orientées vers la performance.

Critères de validation :

- Tous les niveaux de cache disponibles dans le CMS sont activés et fonctionnent correctement.
- Les performances du site, mesurées par des outils de test de vitesse, montrent une amélioration significative après l'activation du cache.
- Les ressources serveur utilisées (CPU, mémoire) sont réduites par rapport à une configuration sans cache.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 70

AR-HE-01 – Héberger toutes les ressources statiques sur un même domaine et les placer sur un CDN

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Fondamentaux de l'hébergement
Complexité : 3 – Priorité : 3

En hébergeant toutes les ressources statiques (comme les images, les fichiers JavaScript, les feuilles de style CSS, etc.) sur un seul domaine, vous réduisez le nombre de requêtes HTTP nécessaires. Encore plus efficace, si ce serveur est un CDN (Content Delivery Network), nous rapprochons physiquement les fichiers/ressources des utilisateurs dans le cas où notre application est utilisée dans plusieurs régions géographiques, ce qui réduit la charge réseau et améliore le temps de réponse.

Attention cependant à ne pas systématiser cette approche. Si votre service est destiné à une audience très localisée, un hébergement local pourrait être plus efficace d'un point de vue du nombre de ressources à solliciter que de mettre en place un CDN en plus du serveur applicatif.

Moyens de mise en œuvre :

- Activer HTTP/2 ou HTTP/3 sur votre serveur. Ces versions du protocole HTTP permettent le multiplexage, ce qui signifie que plusieurs requêtes peuvent être envoyées en même temps sur une seule connexion TCP. Cela réduit le temps de chargement des pages.
- Limiter le nombre de domaines différents pour les ressources utilisées. En hébergeant toutes les ressources statiques sur un seul domaine, on réduit le nombre de requêtes DNS.
- Utiliser des CDN pour héberger les bibliothèques JavaScript, feuilles de style CSS, et autres fichiers statiques. Par exemple, les CDN fournis par Google hébergent de nombreuses bibliothèques JavaScript couramment utilisées.
- Évaluer la pertinence de l'utilisation d'un CDN en fonction de l'audience et du trafic. Si l'audience est très localisée ou si le site a peu de trafic, un hébergement local pourrait être plus approprié.

Critères de validation :

- L'ensemble de vos ressources prennent en charge HTTP/2 ou HTTP/3.
- Les ressources sont servies par un maximum de 5 domaines différents.
- Plus de 50% des ressources sont servies par un CDN (si cette option est pertinente au regard de votre contexte).

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=100
- Les 115 bonnes pratiques d'écoconception web : 69
- Les 115 bonnes pratiques d'écoconception web : 81

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Fondamentaux de l'hébergement
Complexité : 3 – Priorité : 3

Il s'agit de s'assurer que les éléments statiques de votre site web, comme les images, les fichiers CSS et les scripts JavaScript, soient hébergés sur un domaine qui ne nécessite pas l'utilisation de cookies. A chaque fois qu'un navigateur demande une ressource, il envoie également les cookies associés au domaine, même si ces cookies ne sont pas nécessaires pour ces types de contenus. Cela alourdi les requêtes HTTP inutilement. En hébergeant ces ressources sur un domaine spécifique, par exemple static.mondomaine.com, vous évitez d'envoyer des cookies inutiles, ce qui réduit la charge sur le réseau.

Moyens de mise en œuvre :

De nombreux leaders du web ont déjà adopté cette approche. Yahoo! utilise le domaine yimg.com pour ses images, YouTube a son domaine ytimg.com pour les contenus statiques. En suivant leur exemple, vous pouvez configurer votre serveur pour servir les ressources statiques à partir d'un domaine distinct, ce qui vous permettra de bénéficier des mêmes avantages en termes de performance. Cela peut impliquer des ajustements dans votre configuration DNS et votre serveur web.

Critères de validation :

- Les ressources statiques sont effectivement servies depuis un domaine distinct sans cookies.
- Les performances du site, mesurées par le temps de chargement des ressources, montrent une amélioration par rapport à une configuration où les ressources statiques seraient servies avec des cookies.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 80

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Fondamentaux de l'hébergement
Complexité : 3 – Priorité : 2

L'utilisation de solutions de cloud computing se présente comme une option particulièrement avantageuse pour stocker des données sur des infrastructures mutualisées proposant un type de stockage adapté au type de données. En optant pour le cloud, non seulement vous optimisez l'espace de stockage nécessaire pour les ressources de votre service, mais vous bénéficiez également d'une solution évolutive qui s'adapte à vos besoins. Cela signifie que vous n'avez pas à investir dans des infrastructures physiques coûteuses qui pourraient rester sous-utilisées.

Moyens de mise en œuvre :

Plutôt que d'ajouter des disques durs supplémentaires à vos serveurs existants lorsque le volume de stockage de votre service atteint un plafond, ce qui engendrera des coûts supplémentaires, il est beaucoup plus efficace de recourir à un service de stockage cloud scalable à volonté.

Critères de validation :

- Toutes les ressources statiques, telles que les images et les vidéos, sont hébergées sur un service de cloud.
- Aucune nouvelle infrastructure de stockage physique ne doit être ajoutée après la mise en service initiale du service numérique.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 79

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Fondamentaux de l'hébergement
Complexité : 3 – Priorité : 3

La virtualisation est une approche technologique qui permet de créer des versions virtuelles de plusieurs serveurs sur une même machines physiques. On peut alors allouer des ressources physiques partagées nécessaires (comme la mémoire vive, le processeur et l'espace de stockage) à différents environnements virtuels en fonction de leurs besoins spécifiques. Cette méthode présente plusieurs avantages notables : elle permet de diminuer le nombre de serveurs physiques nécessaires et de maximiser le taux d'occupation des ressources. En optimisant l'utilisation des capacités des serveurs, on réduit le besoin de la fabriquer de nouvelles machines et ultimement la quantité de déchets électroniques générés (DEEE). Une machine virtuelle peut également servir de plateforme pour la conteneurisation, permettant ainsi d'exécuter plusieurs applications de manière isolée sur un même système d'exploitation, ce qui renforce encore l'efficacité des ressources allouées en réduisant le nombre de couches logicielles nécessaires en parallèles pour faire tourner plusieurs services simultanément.

Moyens de mise en œuvre :

- Plutôt que de déployer quatre serveurs dédiés pour quatre projets distincts, chacun étant utilisé à seulement 20 % de sa capacité, utilisez des outils de virtualisation reconnus tels que VMware, Xen ou KVM, qui permettent de gérer efficacement les environnements virtuels.
- Intégrer des solutions de conteneurisation comme Docker ou Kubernetes, qui facilitent le déploiement et la gestion d'applications dans des conteneurs légers.

Critères de validation :

- Tous les services sont exécutés sur des machines virtuelles ou des conteneurs, garantissant ainsi une utilisation optimale des ressources.
- La quantité de machines nécessaires pour faire fonctionner plusieurs services est minimisée.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 91

AR-HE-05 – Choisir un hébergeur prouvant une démarche de réduction de son empreinte environnementale

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Choix de l'hébergeur
Complexité : 2 – Priorité : 3

Dans le paysage numérique, les centres de données jouent un rôle central, mais ils sont également responsables d'environ 15 % de l'empreinte carbone du secteur numérique en France (cet impact est en train d'exploser à cause de la généralisation de l'intelligence artificielle générative). Leur impact ne se limite pas seulement aux émissions de gaz à effet de serre (GES), mais englobe également d'autres aspects environnementaux. Il est donc important de choisir un hébergeur qui reconnaît son empreinte écologique et s'engage activement à la réduire. Notre but est de favoriser des solutions d'hébergement qui adoptent une approche proactive en matière de durabilité. Cela inclut la transparence sur leurs performances environnementales et des engagements concrets pour minimiser leur impact.

Moyens de mise en œuvre :

- Choisir un hébergeur transparent : Optez pour un fournisseur qui communique clairement sur son empreinte environnementale et ses engagements concrets pour la réduire. Par exemple :
 - OVH : <https://corporate.ovhcloud.com/fr/sustainability/environnement/>
 - Scaleway : <https://www.scaleway.com/fr/leadership-environnemental/>
 - Infomaniak : <https://www.infomaniak.com/fr/ecologie>
- Analyser les indicateurs environnementaux : Une analyse du cycle de vie (ACV) est idéale, mais au minimum l'hébergeur doit fournir des données sur les émissions de GES (à la fois market-based et location-based), ainsi que sur sa consommation d'énergie, d'eau et de ressources abiotiques (comme les minéraux et les métaux).
- Vérifier les engagements : Assurez-vous que l'hébergeur a pris des engagements sérieux pour réduire son empreinte environnementale, en se basant sur des normes reconnues telles que :
 - Le Code de conduite européen sur les centres de données, établi par la Commission européenne.
 - Les normes environnementales comme ISO 14001.
 - Les certifications environnementales telles que LEED, BREEAM, HQE, etc.
 - L'évaluation des impacts via des référentiels reconnus (ex. : Bilan Carbone, GHG Protocol, Analyse de Cycle de Vie etc.).
- Évaluer les pratiques techniques : Examinez les méthodes de refroidissement utilisées pour minimiser la consommation d'énergie et d'eau, ainsi que les mécanismes de récupération de la chaleur fatale.
- Vérifier la politique de construction durable : Vérifiez que l'hébergeur a une politique de construction qui minimise l'artificialisation des sols et qui respecte des normes de durabilité.
- Rester vigilant face au greenwashing : Soyez attentif aux communications qui semblent trop belles pour être vraies, notamment en ce qui concerne l'atteinte de la neutralité carbone (notion qui n'est pertinente qu'à l'échelle de la planète et pas d'un acteur isolé), et assurez-vous qu'il existe des preuves tangibles de l'engagement environnemental.

Critères de validation :

- L'hébergeur fournit des données claires et vérifiables sur son empreinte environnementale, y compris des certificats relatifs à ses performances environnementales.
- Les méthodologies utilisées par l'hébergeur pour mesurer son impact environnemental sont basées sur des normes et standards reconnus (comme ISO, etc.).
- Les engagements de l'hébergeur pour réduire son empreinte sont documentés et concernent au moins les

indicateurs suivants : émissions de GES (market-based et location-based), consommation d'énergie, d'eau et de ressources abiotiques.

- L'hébergeur a ratifié le Code de conduite sur les centres de données et peut démontrer des actions concrètes en lien avec ce code.
- La déclaration d'écoconception du service inclut des informations sur l'empreinte environnementale de l'hébergement (A minima PUE, WUE et durée de vie moyenne des serveurs).

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=108

AR-HE-06 – Choisir un hébergeur capable de fournir une politique de gestion durable des équipements

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Choix de l'hébergeur

Complexité : 2 – Priorité : 3

L'objectif est de privilégier un hébergement qui affiche des engagements clairs en matière de gestion de ses équipements. Cela inclut une attention particulière aux impacts environnementaux liés à l'acquisition de ces équipements, une politique d'achat qui favorise des produits durables et réparables, une approche réfléchie concernant leur utilisation (comme les mises à jour logicielles, les mises à niveau techniques et les réparations), et enfin une stratégie de fin de vie qui prend en compte le réemploi et la gestion des déchets d'équipements électriques et électroniques (DEEE).

Moyens de mise en œuvre :

Cela peut passer par une demande d'informations précises sur la durée de vie moyenne de ses équipements, tels que les serveurs, les commutateurs réseau, les pare-feux et les routeurs. Nous préconisons qu'un hébergeur s'engage à ce que la durée de vie de ses équipements soit supérieure à huit ans, ou à établir un objectif en ce sens si ses installations sont plus récentes. Il est également recommandé de se renseigner sur la politique d'achat de l'hébergeur, en s'assurant qu'elle privilégie des produits plus durables et réparables, ainsi que sur les pratiques mises en place pour prolonger la vie des équipements.

Critères de validation :

- Un plan de gestion durable des équipements informatiques est en place chez l'hébergeur.
- Ce plan inclut des informations détaillées sur la durée de vie des équipements utilisés.
- La politique d'achat durable de l'hébergeur est clairement définie et accessible.
- Des actions concrètes sont mises en œuvre pour minimiser l'empreinte environnementale tout au long du cycle de vie des équipements.
- Ce plan de gestion durable est référencé dans la déclaration d'écoconception du service numérique proposé.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=110

AR-HE-07 – Choisir un hébergeur avec le PUE (Power Usage Effectiveness) le plus faible possible

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Choix de l'hébergeur
Complexité : 2 – Priorité : 3

Notre engagement est de sélectionner des hébergeurs garantissant une performance énergétique optimale. Pour cela, privilégiez les hébergeurs dont le Power Usage Effectiveness (PUE) est le plus bas possible, à service rendu égal. Le PUE est un indicateur qui mesure l'efficacité énergétique d'un centre de données. S'il est proche de 1, cela signifie que la quasi-totalité de l'énergie utilisée par le centre de données sert à alimenter les serveurs. En plus du PUE, il est important de choisir des hébergeurs qui utilisent des serveurs économes en énergie.

Moyens de mise en œuvre :

Commencez par rechercher des hébergeurs qui partagent leur politique environnementale et diffusent la valeur de leurs PUE.

Pour les hébergements en activité depuis plus de deux ans, le PUE doit être mesuré sur leur valeur dite « réelle » et non « by design », en se référant à des normes et standards internationalement reconnus (par exemple, les normes ISO). Sélectionnez un hébergeur qui indique son PUE et la stratégie mise en œuvre pour le réduire. La communication de cet indicateur doit s'accompagner de la méthodologie utilisée pour le mesurer, basée sur des standards internationalement reconnus (par exemple, ISO/IEC 30134) et notamment sur le type de PUE qui est mesuré (type 1, 2 ou 3). Le PUE est un ratio entre l'énergie totale consommée par l'ensemble du centre d'exploitation (incluant le refroidissement, le traitement d'air, les onduleurs, etc.) et l'énergie effectivement consommée par les systèmes informatiques (serveurs, stockage, réseau). Un PUE proche de 1 indique une excellente performance énergétique. En général, un PUE de 1,1 est constaté pour les hyperscalers et de 2 pour les centres de données plus anciens. Cependant, améliorer cet indicateur peut dégrader d'autres indicateurs, sans réduire l'impact global ni la consommation d'énergie ; d'où l'intérêt de suivre plusieurs indicateurs (consommation d'énergie, consommation d'eau, politique de gestion du matériel, etc.). À noter : Les centres de données les plus récents n'ont pas encore le taux de remplissage suffisant pour que le PUE réel soit l'indicateur approprié. Ainsi, pour les centres de données en activité depuis moins de deux ans, le PUE by design est un meilleur indicateur auquel se référer, mais uniquement pour éviter les risques d'obsolescence.

Critères de validation :

- Le PUE de l'hébergeur du service numérique est vérifié et publié.
- Le PUE de l'hébergeur est inférieur à 1,5 en réel (ou un PUE by design inférieur ou égal à 1,3 si les installations de l'hébergeur sont entrées en activité depuis moins de deux ans).
- Préciser et fournir un lien prouvant le PUE de l'hébergement dans la déclaration d'écoconception du service.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 95
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=111

AR-HE-08 – Choisir un hébergement utilisant une électricité à faible impact environnemental

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Choix de l'hébergeur
Complexité : 2 – Priorité : 3

En plus du PUE, il convient de surveiller l'intensité carbone du pays où se situe le centre de données, et que ce pays soit idéalement dans une région proche de la majorité de vos utilisateurs. Bien qu'il n'existe pas encore de mécanisme parfait pour garantir l'origine de l'électricité consommée, vous pouvez opter pour des centres de données avec un mixe électrique moyen faible en carbone.

Moyens de mise en œuvre :

Il est préférable de choisir un hébergeur dont les serveurs sont situés dans un pays avec une électricité bas carbone. Des outils comme Electricity Maps permettent d'identifier ces pays. En plus de cela, choisir un hébergeur proche des utilisateurs ou des activités principales. Cela ne signifie pas nécessairement adopter des technologies comme l'Edge Computing, mais simplement choisir un centre de données situé à proximité des utilisateurs pour réduire la distance parcourue par les données et, par conséquent, l'infrastructure réseau mobilisée et son empreinte environnementale.

Critères de validation :

- La localisation des utilisateurs et celle des serveurs de l'hébergement sont identifiées.
- L'hébergeur du service numérique est situé dans un pays où l'intensité carbone de la consommation électrique est inférieure à 100 gCO₂eq/kWh, conformément à la trajectoire de réduction des émissions de gaz à effet de serre définie par l'initiative Science-Based Targets (SBTi) et les objectifs de l'Accord de Paris. À partir de 2030, ce seuil doit être de 80 gCO₂eq/kWh, puis 0 gCO₂eq/kWh à partir de 2050.
- La documentation dans la déclaration d'écoconception inclut la localisation (pays, ville) de l'hébergement du service numérique.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=116
- Les 115 bonnes pratiques d'écoconception web : 94
- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=15

AR-HE-09 – Choisir un hébergement qui traite efficacement la chaleur produite par les serveurs

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Choix de l'hébergeur

Complexité : 3 – Priorité : 2

L'objectif est de promouvoir des solutions qui permettent de réduire la consommation d'énergie, et de valoriser la chaleur générée par les serveurs. En effet, cette chaleur, souvent considérée comme un déchet, peut être récupérée et utilisée, par exemple pour chauffer des bureaux ou d'autres bâtiments à proximité pendant les mois froids. Pour mieux comprendre ce concept, se référer à la notion de chaleur fatale, telle que définie par l'ADEME, qui désigne la chaleur non utilisée produite par des installations industrielles ou des centres de données.

Moyens de mise en œuvre :

Choisir un hébergeur qui s'engage activement dans la récupération ou la réutilisation de la chaleur fatale générée par ses installations, par exemple via l'utilisation de cette chaleur pour le chauffage des bureaux ou d'autres infrastructures à proximité.

Critères de validation :

- L'hébergeur a mis en place des initiatives documentées qui assurent la récupération et la réutilisation de la chaleur fatale produite par son installation.
- Le bilan environnemental global de cette réutilisation de la chaleur est positif, en prenant en compte l'investissement initial pour la construction ou l'adaptation des installations.
- Ce critère peut être validé si le centre de données utilisé pour le service affiche un PUE (Power Usage Effectiveness) inférieur à 1,3 en réel.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=117

AR-HE-10 – Choisir un hébergeur avec le WUE (Water Usage Effectiveness) le plus faible possible

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Choix de l'hébergeur
Complexité : 2 – Priorité : 2

Indicateur souvent négligé dans le choix d'un hébergement pour les services numériques, le WUE, qui mesure l'efficacité de l'utilisation de l'eau dans les centres de données, est pourtant un indicateur important à suivre pour réduire la consommation d'eau nécessaire au refroidissement des serveurs. Dans un monde où les changements climatiques apportent une tension particulière sur cette ressource critique aux écosystèmes et aux communautés locales, une gestion responsable de l'eau devient essentielle pour éviter les phénomènes de stress hydriques, c'est-à-dire les pénuries d'eau potable. Il est pertinent de considérer le niveau stress hydrique dans la région où se situe l'hébergement : un WUE élevé dans une zone où l'eau est abondante comme dans le nord de la France ou le Canada peut être moins préoccupant que dans une région déjà soumise à des tensions hydriques comme sur le contour méditerranéen.

Moyens de mise en œuvre :

Choisir un hébergeur qui soit transparent sur son WUE pour chaque datacenter et qui explique clairement la méthodologie utilisée pour le mesurer. Cette méthodologie doit être fondée sur des standards reconnus à l'international, comme ceux établis par l'ISO/IEC 30134. Le WUE est calculé comme un ratio entre la quantité d'eau utilisée et l'énergie totale consommée par le centre de données, exprimé en litres par kilowattheure (L/kWh). Pour les hébergements qui sont en activité depuis plus de deux ans, il est préférable d'obtenir des données sur le WUE basé sur des mesures réelles plutôt que sur des estimations théoriques ("by design").

A noter qu'il existe actuellement peu de données ouvertes concernant le stress hydrique local, ce qui complique l'évaluation de cet indicateur. Il faut également garder à l'esprit que l'amélioration du WUE peut parfois entraîner une dégradation des autres indicateurs de performance environnemental. Par exemple un centre de données qui utilise un refroidissement par climatisation sera plus efficient en utilisation en eau qu'un centre de données utilisant le refroidissement par watercooling, mais utilisera plus d'électricité pour faire fonctionner les climatiseurs et sera responsable de fuites de gaz refroidissants, puissants gaz à effets de serre. C'est pourquoi il est recommandé de suivre plusieurs indicateurs, tels que la consommation d'énergie, la consommation d'eau, et la politique de gestion des équipements, afin d'avoir une vision globale de l'impact environnemental de l'hébergement.

Critères de validation :

- L'hébergeur du service numérique démontre une démarche proactive pour minimiser sa consommation d'eau, en suivant des indicateurs pertinents, notamment son WUE.
- La méthodologie de calcul du WUE est clairement précisée, ainsi que le type de WUE utilisé (réel ou "by design").
- Le WUE visé est inférieur ou égal à 0,4 L/kWh, avec une préférence pour des données réelles plutôt que des estimations théoriques, surtout pour les centres de données en activité depuis moins de deux ans.
- La valeur du WUE est communiquée de manière transparente dans la déclaration d'écoconception du service.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=113

AR-HE-11 – Mettre en place une architecture scalable

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Optimisation et architecture
Complexité : 3 – Priorité : 3

L'idée ici est d'éviter de surdimensionner nos infrastructures numériques lorsque cela n'est pas nécessaire. On souhaite une architecture capable d'ajuster dynamiquement les ressources en fonction de la demande. Si nos produits sont utilisés par très peu d'utilisateurs la nuit, il est potentiellement inutile (et coûteux) de maintenir la même quantité de ressources en tout temps. Une architecture élastique permet au contraire d'économiser des coûts et d'optimiser l'efficacité énergétique en allouant les ressources strictement nécessaires. Cela évite le gaspillage et permet de mieux gérer les pics de charge sans surprovisionner.

Moyens de mise en œuvre :

- Commencer par évaluer précisément le nombre d'utilisateurs en fonction des périodes et les besoins en ressources associés nécessaires pour absorber cette charge.
- Utiliser des serveurs virtualisés, notamment sur le cloud, pour pouvoir modifier dynamiquement et automatiquement la taille de l'infrastructure en fonction de la charge. Par exemple, vous pouvez programmer des réductions de ressources la nuit ou en fonction du nombre de requêtes.
- Adopter des solutions comme Docker EE ou Kubernetes pour ajuster les ressources allouées aux services en fonction des variations saisonnières ou des pics d'activité.
- Mettre en place des scripts et des outils d'automatisation pour gérer l'allocation et la désallocation des ressources en temps réel. Cela peut inclure des mécanismes d'auto-scaling qui créent ou suppriment des instances en fonction de la demande.

Critères de validation :

- Le ratio entre les ressources allouées et consommées est monitoré. Des outils de surveillance collectent des données sur l'utilisation du processeur, de la mémoire, de la bande passante, etc.
- Des tests de charge ont été effectués pour vérifier si l'architecture peut détecter automatiquement une augmentation de la demande.
- Des mécanismes d'auto-scaling sont mis en place et se déclenchent automatiquement en fonction des conditions de charge.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=52
- Les 115 bonnes pratiques d'écoconception web : 27
- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=17

AR-HE-12 – Mettre en place un système de stockage des données « chaudes » et « froides »

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Optimisation et architecture
Complexité : 3 – Priorité : 1

Les données « chaudes » sont celles que nous utilisons fréquemment, nécessitant un accès rapide, donc celles qui vont solliciter le plus de ressources matérielles et énergétiques. En revanche, les données « froides » sont souvent des données d'archives, moins sollicitées et peuvent être stockées de manière plus économique en matière et en énergie.

De plus, les systèmes de stockage de données « froides » sont généralement moins onéreux aussi longtemps que les opérations de lectures sont réduites à leurs minimum. L'objectif donc est de mettre en place une stratégie de stockage qui respecte cette distinction.

Moyens de mise en œuvre :

Commencer par une analyse des types de données que vous gérez. Identifiez clairement ce qui constitue des données « chaudes » et « froides » dans votre contexte spécifique. Ensuite, choisissez des solutions techniques adaptées pour le stockage de ces données. Par exemple, pour les données « chaudes », optez pour des systèmes de stockage à haute performance qui garantissent un accès rapide (sur SSD par exemple). Pour les données « froides », explorez les options de stockage à long terme, comme les systèmes de stockage en cloud optimisés pour l'archivage (voir les systèmes coldline ou glacier des hébergeurs cloud).

Critères de validation :

- Une séparation claire des données « chaudes » et « froides » est visible dans l'architecture du service numérique.
- Les configurations des systèmes de stockage montrent que des solutions distinctes sont mises en place pour chaque type de données.
- Des stratégies de stockage appropriées pour les données « froides » sont documentées et mises en œuvre.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=118

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Optimisation et architecture

Complexité : 2 – Priorité : 2

On souhaite réduire au maximum l'utilisation des ressources de réseau et de stockage, tout en s'assurant la résilience du service à la perte temporaire ou définitive de données. Poser la question de la pertinence du niveau de service choisi par rapport aux exigences et à la criticité des données est primordial en ce sens. Plus le taux de garantie de disponibilité exigé est élevé, plus cela nécessitera la mise en place d'une infrastructure lourde et redondante. Il s'agit de trouver un juste équilibre entre la sécurisation des données, pour éviter toute perte, et la dissémination excessive, qui pourrait entraîner une redondance et des flux inutile.

Moyens de mise en œuvre :

Identifier les données critiques qui nécessitent réellement une duplication, puis choisir l'approche à implémenter :

- Backup & Restore : Pour les applications qui peuvent se permettre un temps d'arrêt de quelques heures, avec un RTO (Recovery Time Objective) ou RPO (Recovery Point Objective) adapté. Il faudra s'assurer de stocker le Backup sur une autre infrastructure que le serveur hébergeant les données, à la fois physiquement sur un autre serveur, et géographiquement dans un autre datacenter.
- Pilot Light : Cette méthode consiste à avoir une base de données « miroir » ou dupliquée, mais avec des machines virtuelles éteintes quand les opérations de duplication ne sont pas en marche. Bien que ce soit un peu plus coûteux qu'un simple Backup & Restore, cela convient à la plupart des applications qui n'ont pas d'exigences SLA (Service Level Agreement) très strictes, c'est-à-dire inférieures à une heure.
- Warm Standby : Ici, les machines virtuelles miroirs sont déjà en fonctionnement, mais avec une scalabilité limitée. Cela permet une quasi-continuité de service, bien que la qualité puisse être légèrement dégradée en cas d'incident le temps que les machines se dimensionnent correctement par rapport à la charge.
- Hot Standby multisite : Cette approche offre une résilience totale pour des SLAs en temps réel, où aucune perte de service n'est tolérée. Cependant, cela implique nécessairement des coûts plus élevés et une grande redondance des ressources allouées.

Critères de validation :

- Un SLA (Service Level Agreement) est présent et ajusté en fonction des besoins spécifiques du service.
- Les données dupliquées sont clairement identifiées et justifiées par leur criticité ou leur fréquence d'utilisation.
- Les méthodes de sauvegarde et de restauration choisies sont en adéquation avec les exigences de disponibilité et de performance du service.
- Une évaluation régulière est effectuée pour s'assurer que la stratégie de duplication reste pertinente et efficace face à l'évolution des besoins.
- Bonus : on peut également définir un SLA par fonctionnalité, afin de garantir un haut niveau de disponibilité des opérations critiques tout en permettant plus de pannes sur les fonctionnalités satellites à moins forte valeur ajoutée, afin de moduler le besoin de réplication à un périmètre plus restreint du service.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=119

AR-HE-14 – Installer le minimum requis sur le serveur

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Optimisation et architecture

Complexité : 3 – Priorité : 3

Faites le ménage sur votre serveur en désinstallant tous les services qui ne sont pas absolument nécessaires au bon fonctionnement de votre service afin d'alléger la charge et réduire la consommation de ressources, notamment en cycles CPU et en mémoire vive. Vous aurez un serveur plus performant, car moins encombré par des processus inutiles qui tournent en arrière-plan, contribuant à une meilleure efficacité énergétique.

Moyens de mise en œuvre :

Il est conseillé de privilégier une installation manuelle de votre serveur. Évitez les distributions qui viennent avec des surcouches lourdes comme cPanel ou Plesk, car elles incluent souvent de nombreux services par défaut qui ne vous seront pas forcément utiles. Si vous avez besoin d'une interface d'administration, tournez-vous vers des solutions plus légères comme Webmin, qui vous permettront de gérer votre serveur sans alourdir son fonctionnement.

Pour les services containerisés, il est préférable de choisir des images de type "slim" ou "alpine". La version 22 de l'image Node sur Dockerhub fait respectivement 385.61 MB pour la version par défaut, 73.88 MB pour la version slim et 50.91 MB pour la version Alpine. Il convient de toujours construire ses images Docker en multi-stage build, qui permet de débarrasser l'image finale de tous les fichiers sources et les packages nécessaires à la compilation qui ne seront plus nécessaires en phase de run.

Critères de validation :

- Le serveur a été nettoyé de tous services non nécessaires au fonctionnement du service exécuté.
- Un audit régulier est effectué pour s'assurer qu'aucun service superflu n'a été ajouté au fil du temps.
- Les images de container se basent sur des versions optimisées.
- Les container sont construits avec du multi-stage.
- La performance du serveur est mesurée avant et après la désinstallation des services pour évaluer l'impact de cette démarche.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 93

AR-HE-15 – Optimiser la gestion des environnements de développement, de préproduction et de test en fonction de ses besoins

Ateliers d'architecture logiciel, devops, hébergement / Hébergement / Optimisation et architecture
Complexité : 2 – Priorité : 2

Faire fonctionner des serveurs ou des machines virtuelles qui ne sont pas utilisés représente une perte de ressources, tant sur le plan matériel, énergétique et économique. L'objectif de cette bonne pratique est donc de réduire cette sollicitation inutile de ressources en rationnant l'utilisation des environnements de développement, de préproduction et de test. Cela peut se faire en mutualisant ces environnements ou en les éteignant durant les périodes où ils ne sont pas nécessaires, comme la nuit ou pendant les week-ends. Libérer du matériel non sollicité permet de le réallouer à d'autres usages, particulièrement dans des centres de données mutualisés comme chez les hyperscalers.

Moyens de mise en œuvre :

Tout d'abord, il est recommandé de mettre en place des environnements de développement, de préproduction ou de test sur des infrastructures mutualisées, comme par exemple des machines virtuelles partagées. Pour ce type d'environnements, les contraintes de sécurité de production qui peuvent parfois surpasser la possibilité de mutualisation des environnements peut être réévaluer au regard de la criticité des données hors production.

Si la mutualisation n'est pas une option viable, il est alors conseillé de désactiver ces environnements durant les plages horaires où ils ne sont pas utilisés. Par exemple, on peut programmer l'extinction des serveurs la nuit et les remettre en service automatiquement à une heure précise, ou bien en fonction d'un signal indiquant qu'ils pourraient être nécessaires prochainement. Cette approche permet de garantir que les ressources ne sont sollicitées que lorsque cela est réellement nécessaire, tout en offrant une flexibilité pour les équipes de développement.

Critères de validation :

- Le service utilise des environnements de développement, de préproduction ou de test mutualisé.
- Les environnements sont désactivés durant les périodes où ils ne sont pas utilisés.
- Des indicateurs de performance sont mis en place pour suivre l'utilisation des ressources et s'assurer que les pratiques d'optimisation sont respectées.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=59

AR-IA-01 – Choisir un modèle d'IA proposant un niveau de complexité minimisé et proportionné au besoin effectif du service

*Ateliers d'architecture logiciel, devops, hébergement / Intelligence Artificielle / Choix du modèle
Complexité : 2 – Priorité : 3*

S'il est nécessaire de recourir à un modèle d'apprentissage, il s'agit de sélectionner une approche d'entraînement frugale et appropriée, en tenant compte de l'usage attendu du service. Certains modèles d'IA sont moins énergivores et ont besoin de moins de données en entrée que d'autres modèles.

Moyens de mise en œuvre :

Privilégier des méthodes d'apprentissage qui soient simples et peu coûteuses en termes de ressources. Par exemple, on peut opter pour des techniques de régression, qui sont généralement moins exigeantes, ou, si nécessaire, des réseaux de neurones simples. Il est préférable d'éviter les technologies d'apprentissage profond (deep learning), qui, bien qu'efficaces dans certains cas, consomment beaucoup plus de ressources de calcul et d'énergie. Si un modèle d'apprentissage profond est nécessaire, comme par exemple un Large Language Model (LLM), choisir un modèle reposant sur une plus faible quantité de layers et de paramètres permettra de limiter le besoin en ressources pour le faire fonctionner.

Critères de validation :

- La consommation énergétique et les ressources matérielles de la méthode d'entraînement utilisée sont mesurées et reconnues comme étant faibles.
- Le modèle choisi est celui permettant d'utiliser le moins de ressources pour délivrer le besoin attendu.
- Si la méthode choisie n'est pas une régression ou une autre méthode peu coûteuse, il est impératif de justifier ce choix dans la déclaration d'écoconception du service. Cette justification doit démontrer la nécessité d'utiliser des méthodes plus complexes pour le cas d'usage spécifique.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=125
- Référentiel Général pour l'IA Frugale : BP30

AR-IA-02 – Choisir un modèle d'IA open source, publiant des données relatives à l'impact de la phase d'entraînement

Ateliers d'architecture logiciel, devops, hébergement / Intelligence Artificielle / Choix du modèle
Complexité : 3 – Priorité : 3

De nombreuses plateformes de modèles en Open Source comme HuggingFace existent et permettent de facilement trouver des modèles déjà entraînés sur des tâches spécifiques.

En utilisant des modèles déjà entraînés, on évite au maximum des impacts supplémentaires liés aux réentraînements de modèles similaires et au fine-tuning redondants. De plus, des données ouvertes concernant l'impact des modèles prouvent que des efforts quant à l'optimisation de ces impacts sont considérés par le fournisseur, tout en permettant de les intégrer dans le calcul de l'empreinte environnementale de votre service.

Moyens de mise en œuvre :

Choisir des modèles open source proposés sur des plateformes comme HuggingFace (<https://huggingface.co/models?license=license:apache-2.0&sort=trending>).

Certaines entreprises publient ouvertement les données d'impact de l'entraînement de leurs modèles comme le nombre d'heures GPU, la quantité d'électricité consommée et la quantité de GES émis (https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md).

A ses informations, il faudrait idéalement ajouter le nombre et le type de GPU utilisés et la localisation du datacenter dans lequel l'entraînement a eu lieu. Nous encourageons également à considérer l'implication des pays qui hébergent ces entreprises créatrices de modèles d'IA aux efforts collectifs à la transition environnementale, notamment leur adhésion aux accords de Paris.

Critères de validation :

- Le modèle choisi est publié sous licence open source comme Apache2 ou GPL.
- Le modèle rend accessible les données d'impact de sa phase d'entraînement.
- Les données d'impact sont réutilisées dans le calcul d'impact du service numérique.

Liens :

- Référentiel Général pour l'IA Frugale : BP29

Utiliser des modèles ayant été compressé pour réduire leur complexité tout en maintenant leur performance, afin d’alléger la charge sur les serveurs, réduire la quantité de ressources nécessaires à la phase d’inférence et diminuer la consommation d’énergie nécessaire.

Moyens de mise en œuvre :

Plusieurs techniques de compression existent et peuvent être combinées :

- Sparsification : Cette technique consiste à rendre le modèle plus léger en éliminant les poids qui ont peu ou pas d’impact sur les résultats.
- Pruning : Cela implique de couper les connexions inutiles dans le réseau de neurones, ce qui réduit le nombre de paramètres à traiter.
- Unification : Cette méthode regroupe des poids similaires pour réduire la redondance.
- Local scaling : En limitant l’impact de la quantification, cette technique aide à maintenir la précision tout en réduisant la taille du modèle.
- Batch norm folding : Cela permet de réduire la redondance de certains paramètres en intégrant les normalisations dans les couches précédentes.
- Quantification : Cette méthode consiste à réduire la précision des poids, ce qui diminue la taille du modèle sans trop affecter la performance.
- Distillation : Un modèle plus petit est entraîné pour imiter le comportement d’un modèle plus grand, ce qui permet de conserver l’essentiel des performances tout en réduisant la taille.

Critères de validation :

- Une déclaration d’écoconception est fournie, détaillant les techniques mises en œuvre.
- Des comparaisons avant/après sont réalisées pour illustrer les améliorations apportées grâce à la compression des modèles.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=129
- Référentiel Général pour l’IA Frugale : BP03

Ateliers d'architecture logiciel, devops, hébergement / Intelligence Artificielle / Choix du modèle
Complexité : 2 – Priorité : 2

Établir un référentiel des modèles les plus adaptés à vos services accompagnés de leurs impacts environnementaux (prédictifs puis réellement constatés) afin de se familiariser avec les ordres de grandeur et établir des comparaisons d'un projet à un autre.

Moyens de mise en œuvre :

Construire une base de données de modèles open source ou privés comportant les données relatives à l'impact de l'entraînement et les impacts en production de l'inférence (quantité de ressources mobilisées, nombre de requêtes, nombre de token générés, consommation énergétique) afin de suivre dans le temps l'évolution de l'intensité des impacts environnementaux de votre utilisation de l'IA.

Alimenter ce référentiel de façon anticipée puis réelle via différents outils. Alimenter ce référentiel avec l'estimation que fournissent les outils (Code Carbon, Green Algorithms, Carbon Tracker, MLCO2Impact ..), puis la réelle consommation via de PDU. Cela peut permettre de comparer l'estimation et la réelle consommation, ainsi que d'affiner les prédictions des futurs entraînements.

Critères de validation :

- Une base de données d'impact existe concernant les systèmes d'IA utilisés dans les services.

Liens :

- Référentiel Général pour l'IA Frugale : BP21

AR-IA-05 – Utiliser une stratégie d'inférence optimisée en termes de consommation de ressources et des cibles utilisatrices sur toute la vie du projet

Ateliers d'architecture logiciel, devops, hébergement / Intelligence Artificielle / Stratégie d'inférence
Complexité : 1 – Priorité : 3

Traditionnellement, l'on pensait que la phase d'entraînement des modèles était la plus énergivore, mais avec l'essor de l'IA générative, la phase d'inférence devient de plus en plus gourmande en énergie par effet de volume. En effet, le nombre croissant de requêtes des utilisateurs pour des services largement utilisés par le grand public accentue cette consommation. Ainsi, il est important de développer des stratégies qui permettent de rendre cette phase d'inférence plus efficace et moins consommatrice de ressources. Il est possible que le service d'IA connaisse un fort engouement à son lancement et une utilisation au long cours beaucoup plus modeste, ou l'inverse. Afin d'éviter des dépenses et impacts inutiles, il est vital de penser correctement ces différentes phases dès la conception, allant jusqu'à la fin de vie du système avec des scénarios de décommissionnement à la fin du projet et de revalorisation des équipements utilisés.

Moyens de mise en œuvre :

Monitorer des indicateurs qui permettent de suivre la consommation d'énergie et les ressources de calcul (mesures au plus près de la couche physique) sollicités durant la phase d'inférence en se fixant des objectifs pour les réduire. Calculer l'énergie consommée ramenée à la requête ou au token généré pour comparer à l'évolution de l'usage. Éviter les appels superflus qui pourraient alourdir la charge, en mettant en place par exemple un cache qui remet de délivrer des résultats déjà obtenus précédemment sans resolliciter le modèle.

L'utilisation de processeurs (CPU), de processeurs graphiques (GPU) ou de processeurs de tensor (TPU) optimisés pour ces calculs est recommandée, tout en veillant à ce que leur localisation géographique soit choisie de manière à minimiser l'empreinte environnementale, en privilégiant des zones proposant des mixes électriques à faible intensité carbone, et des centres de données possédant de bons indicateurs de performances environnementaux (voir section Architecture - Hébergement).

Pour le lancement du service, on pourra préférer utiliser des infrastructures mutualisées pour ne pas avoir à acquérir immédiatement du nouveau matériel, ou définir une politique de désallocation de ressources matérielles selon l'utilisation réelle du service.

Critères de validation :

- La stratégie d'inférence mise en place est clairement adaptée aux besoins des utilisateurs, tout en minimisant les ressources nécessaires à son fonctionnement.
- Des indicateurs de suivi de la consommation des ressources sont établis et utilisés pour ajuster la phase d'inférence.
- Le service numérique présente dans sa déclaration d'écoconception des preuves tangibles de l'application de principes d'écoconception dans sa phase d'inférence, en adéquation avec les besoins des utilisateurs.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=130
- Référentiel Général pour l'IA Frugale : BP11
- Référentiel Général pour l'IA Frugale : BP24

AR-IA-06 – Favoriser l’usage de l’équipement déjà existant pour l’entraînement et l’inférence

Ateliers d’architecture logiciel, devops, hébergement / Intelligence Artificielle / Réutilisation du matériel
Complexité : 2 – Priorité : 3

L’impact environnemental de la fabrication d’un équipement correspond à plusieurs années de son utilisation, c’est pourquoi une vigilance toute particulière doit être portée à la réutilisation en priorité de l’existant avant tout achat additionnel.

Pour la phase d’entraînement comme d’inférence, on essaiera toujours d’utiliser en priorité le matériel déjà existant, dans une logique de frugalité contrainte.

Moyens de mise en œuvre :

Mutualiser les équipements existants pour atteindre une utilisation maximale ou une plus haute puissance de calcul sans rachat de matériel.

Mettre le matériel existant sous tension pour challenger le besoin de puissance nécessaire pour réaliser l’entraînement ou l’inférence, que cela soit les ordinateurs des développeurs ou les serveurs.

Critères de validation :

- Des tests de charge ont été rigoureusement effectués pour envisager une mutualisation d’équipement
- Le matériel existant a d’abord été sollicité et le besoin challengé avant d’acquérir du nouveau matériel.

Liens :

- Référentiel Général pour l’IA Frugale : BP20
- Référentiel Général pour l’IA Frugale : BP22

AR-SP-01 – Imposer une démarche d'écoconception à vos fournisseurs

Ateliers d'architecture logiciel, devops, hébergement / Spécifications / Les fournisseurs

Complexité : 3 – Priorité : 3

L'écoconception d'un service numérique ne doit pas se limiter à l'organisation qui produit ce service, mais doit également inclure une responsabilité environnementale qui s'étend à l'ensemble de la chaîne de valeur, y compris les fournisseurs. Il est fondamental de s'assurer que ces fournisseurs adoptent des pratiques visant à réduire leur impact environnemental. Cela permet non seulement de minimiser l'empreinte de votre service numérique, mais aussi de promouvoir une culture de durabilité au sein de l'ensemble de l'écosystème de production car effet dominos.

Moyens de mise en œuvre :

Commencer par identifier toutes les ressources nécessaires au projet, qu'elles soient internes ou externes. Une fois ces ressources identifiées, il convient de leur associer des exigences environnementales claires et précises. Il est important de noter que le périmètre de cette démarche se concentre sur la conception du service numérique, sans s'étendre aux pratiques internes des fournisseurs eux-mêmes. Pour guider cette démarche, il est recommandé de se référer à des documents tels que le "Guide pratique pour des achats numériques responsables", qui offre des conseils pratiques, ainsi qu'à la norme ISO 20400, qui fournit des lignes directrices sur les achats responsables et sur la manière d'intégrer des objectifs de responsabilité sociétale tout au long de la chaîne d'approvisionnement.

Critères de validation :

- Les caractéristiques environnementales des fournisseurs sont clairement prises en compte dans la politique d'achat ou de partenariat du service.
- Les impacts environnementaux associés aux choix des fournisseurs sont documentés et intégrés dans la déclaration d'écoconception.
- Les recommandations du "Guide pratique pour des achats numériques responsables" et/ou de la norme ISO 20400 sont effectivement appliquées et respectées dans le cadre des relations avec les fournisseurs.
- Une évaluation régulière des pratiques des fournisseurs est effectuée pour s'assurer qu'ils continuent à respecter les exigences environnementales établies.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=46

AR-SP-02 – Sélectionner des composants d'interface ayant été pensé dans une démarche d'écoconception

Ateliers d'architecture logiciel, devops, hébergement / Spécifications / Les composants

Complexité : 3 – Priorité : 1

Les composants d'interface, tels que les boutons, les formulaires et autres éléments graphiques, qui sont souvent intégrés à partir de bibliothèques ou de systèmes de design, peuvent alourdir considérablement une interface au regard du poids des ressources à télécharger et la complexité du DOM, mais aussi compromettre la compatibilité du service avec des versions de navigateurs anciens. Sélectionner et observer les effets de ces composants sur notre service est impératif afin de ne pas ruiner les efforts déployés par la démarche d'écoconception.

Moyens de mise en œuvre :

Procéder à une analyse comparative des différents composants d'interface que nous envisageons d'utiliser. Cela signifie qu'il faut évaluer leur impact sur le service en examinant des critères tels que la taille des fichiers, la quantité de données qu'ils nécessitent pour fonctionner, et leur efficacité en termes de ressources. Par exemple, nous pourrions mesurer le poids des fichiers associés à ces composants et évaluer combien de données sont transférées lors de leur utilisation. Il est également judicieux de se renseigner sur les pratiques de conception et les efforts déployés par l'équipe qui a conçu ces composants en termes d'optimisation des ressources.

Critères de validation :

- La majorité des composants d'interface utilisés dans le service sont conçus pour minimiser leur impact sur le service dans lequel ils sont implémentés.
- Des mesures comparatives ont été effectuées entre les différents composants similaires, et ceux qui présentent les meilleures performances ont été sélectionnés.
- Des techniques telles que l'utilisation de méthodes de compression efficaces, l'optimisation des ressources, et la minimisation des transferts de données ont été prises en compte dans le processus de sélection.
- Les choix effectués sont documentés dans une déclaration d'écoconception, précisant les raisons de ces choix et les impacts estimés.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=47

AR-SP-03 – Sélectionner des services tiers ayant été pensé dans une démarche d'écoconception

Ateliers d'architecture logiciel, devops, hébergement / Spécifications / Les services tiers

Complexité : 3 – Priorité : 3

Les services tiers sont des solutions fournies par des entreprises ou des développeurs externes qui proposent des fonctionnalités prêtes à l'emploi, comme l'envoi de mails, la génération de contenus grâce à des modèles d'apprentissage ou de la recherche. Le niveau d'abstraction de ces services nous impose de nous assurer qu'ils ne nuisent pas à notre engagement envers la sobriété numérique et l'écoconception de nos produits. L'objectif est de réduire les impacts environnementaux associés à l'utilisation de ces services externes, en prenant des décisions éclairées basées sur des critères environnementaux lors de leur sélection.

Moyens de mise en œuvre :

Les détails techniques d'une solution tier étant toujours opaque, les éléments sur lesquels se baser sont :

- Le service tier donne-t-il des informations sur le lieu et le type d'hébergement ? Si oui, appliquer les mêmes critères de sélection qu'en interne.
- Le service tier revendique-t-il de mettre en place une démarche de réduction de ses impacts ? Si oui, c'est un bon début et si faut investiguer.
- Le service tier fournit-il des données relatives à ses impacts comme un bilan environnemental ? Si oui, l'intégrer à notre propre bilan.

Critères de validation :

- Une déclaration d'écoconception est fournie, incluant une liste des services tiers utilisés et leur avancement par rapport aux critères environnementaux, notamment :
 1. Le service communique-t-il une déclaration d'écoconception ?
 2. Le service communique-t-il les mesures d'impact de l'utilisation de ses systèmes ?
 3. Le service est-il engagé dans une démarche de réduction de ses impacts ?

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=48

AR-ST-01 – Collecter les données de façon responsable et raisonnée

Ateliers d'architecture logiciel, devops, hébergement / Stratégie / Collecte des données

Complexité : 2 – Priorité : 2

Cette bonne pratique vise à encourager une collecte de données raisonnable et raisonnée, en allant au-delà des simples obligations légales de minimisation prévues par le RGPD. L'idée est de réduire la quantité de données collectées, traitées et stockées par le service numérique, y compris les données non personnelles et les métadonnées. Il s'agit de collecter uniquement les données essentielles pour améliorer l'expérience utilisateur et le fonctionnement du service.

Moyens de mise en œuvre :

- Identifier clairement les données nécessaires au bon fonctionnement du service. Si une donnée ne contribue pas directement à l'amélioration de l'expérience utilisateur ou au fonctionnement du service, envisager de ne pas la collecter.
- Éviter de collecter des métadonnées dans une perspective de profilage des utilisateurs. Si cette collecte est indispensable, il faut s'assurer qu'elle respecte les dispositifs du RGPD.
- Pour les données non personnelles, définir une durée de conservation minimale afin d'éviter un stockage excessif.
- Mettre en place une collecte de consentement pour les utilisateurs, de préférence en opt-in plutôt que opt-out, incluant le droit de s'opposer et/ou de donner leur consentement explicite pour l'ensemble des données collectées, y compris les données non personnelles.
- Limiter à un seul outil pour suivre les actions des utilisateurs. Éviter d'utiliser plusieurs outils simultanément.

Critères de validation :

- Le type et la quantité de données collectées, traitées et stockées par le service sont justifiés.
- Pour les données personnelles, la minimisation de la collecte est respectée conformément au RGPD.
- La limitation du traitement et de la durée de conservation des données est documentée.
- Les outils de recueil du consentement sont mis en place.
- Le nombre d'outils d'Analytics utilisés est limité à un seul.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=28
- Les 115 bonnes pratiques d'écoconception web : 111

Sprint Planning

Cérémonie au cours de laquelle l'équipe Scrum définit un objectif clair pour un sprint à débiter, sélectionne les US à réaliser et planifie la manière dont elles seront réalisées.

SP-API-01 – Limiter le nombre d'appels aux API HTTP

Sprint Planning / API / Optimisation des performances

Complexité : 3 – Priorité : 3

L'appel à une API HTTP engrange des échanges réseaux et de la charge sur les serveurs qui devront traiter notre requête. Afin de minimiser la sollicitation de ressources inutiles et les impacts associés, il est intéressant d'adopter une approche réfléchie quant à notre consommation d'API. Une stratégie de mise en cache efficace et une réflexion sur la structure des API permet généralement de diminuer drastiquement le nombre d'appels à effectuer pour un acte métier.

Moyens de mise en œuvre :

- Limiter le nombre de requêtes à un certain nombre d'occurrence par jours.
- Utiliser le cache navigateur (localStorage, indexeddb, cache manifest) pour conserver les données pour les réutiliser entre 2 rafraichissements.

Critères de validation :

- Tous les points de terminaison (endpoints) bénéficient d'une stratégie de cache pertinente coté frontend.
- Le nombre de requêtes vers les API respecte le budget environnemental fixé pour le parcours utilisateur défini (voir pratique ECO-ST-03 Établir un budget environnemental)

Liens :

- Les 115 bonnes pratiques d'écoconception web : 21

SP-API-02 – Implémenter et utiliser des filtres pour limiter les champs renvoyés par l'API

Sprint Planning / API / Optimisation des performances

Complexité : 2 – Priorité : 2

Une API peut parfois renvoyer beaucoup plus d'informations que nécessaire. Cela concerne en particulier les endpoint retournant des tableaux d'objets, ou des objets imbriqués sur plusieurs niveaux. Cela se traduit par une sollicitation inutile du réseau en termes de volume de données. En concevant des filtres efficaces qui permettent aux développeurs frontend de restreindre le volume de données demandées, on optimise la quantité de réseau et de stockage en cache à fonctionnalité équivalente.

Moyens de mise en œuvre :

- Pour les concepteurs d'API : Concevoir l'API de manière à inclure des options de filtrage dans les requêtes, permettant aux développeurs de spécifier les champs ou la quantité d'objets qu'ils souhaitent récupérer.
- Pour les consommateurs d'API : Utiliser au maximum les paramètres de requête GET et se demander systématiquement si toute la donnée récupérée sera utile pour les traitements.

Critères de validation :

- Les requêtes API peuvent être exécutées avec des filtres qui limitent les champs ou les quantités retournés.
- Les utilisateurs peuvent spécifier des paramètres de requête pour les méthodes GET.
- Les tests montrent une réduction mesurable du volume de données transférées lors des requêtes utilisant des filtres et des paramètres de requête.
- La documentation de l'API inclut des exemples clairs d'utilisation des filtres et des paramètres de requête pour encourager leur utilisation.

Liens :

- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=24
- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=27

Sprint Planning / API / Optimisation des performances

Complexité : 2 – Priorité : 2

L'implémentation de la pagination des tableaux est essentielle pour gérer efficacement les réponses des API, surtout lorsque celles-ci peuvent renvoyer un volume de données considérable. En limitant la quantité de données retournées, on s'assure que seules les informations nécessaires sont envoyées au service qui consomme la donnée. Cette approche est particulièrement pertinente pour les requêtes qui, sans pagination, pourraient entraîner des temps de réponse longs.

Moyens de mise en œuvre :

- Intégrer des paramètres de pagination dans les requêtes API, permettant aux utilisateurs de spécifier le nombre d'éléments à retourner et de naviguer entre les pages de résultats.
- Documenter clairement l'utilisation de la pagination dans la documentation de l'API afin que les développeurs sachent comment l'implémenter correctement.
- Effectuer des tests de charge pour s'assurer que la pagination fonctionne comme prévu et qu'elle améliore les performances de l'API.

Critères de validation :

- Les requêtes API utilisent les mécanismes de pagination dans les réponses.
- Les temps de réponse de l'API sont améliorés lorsque la pagination est mise en œuvre.
- La documentation de l'API inclut des exemples clairs sur l'utilisation de la pagination.

Liens :

- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=26
- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=30

Sprint Planning / API / Optimisation des performances

Complexité : 2 – Priorité : 3

En utilisant un système de cache serveur pour les données les plus fréquemment utilisées, on améliore le temps de réponse des API, et donc l'expérience utilisateur, et on évite de solliciter les ressources de calcul en évitant d'exécuter plusieurs fois la même requête sur les mêmes données. Il s'agit de placer un cache devant chaque composant de l'architecture qui retourne des données (API, base de données, application frontend, etc.) et en le rapprochant le plus possible des utilisateurs. Les gains attendus sont une réduction de l'utilisation des ressources de calcul et un impact réseau diminué (sur toute la stack technique).

Moyens de mise en œuvre :

- Identifier les données, les entrées API et les ressources les plus souvent utilisées.
- Distinguer le cache côté serveur du cache côté client. Les bottleneck et points de convergence de l'architecture du service sont à prioriser.
- Pour rafraîchir les données en cache, configurer une durée d'expiration cohérente avec le besoin métier associé.
- Implémenter un système pour rafraîchir le cache régulièrement afin de garantir que les données restent à jour. Il est possible d'invalider automatiquement le cache après une période déterminée ou utiliser des mécanismes de purge du cache lorsqu'une mise à jour des données est effectuée.

Critères de validation :

- Les ressources les plus utilisées du serveur sont mises en cache.
- Chaque donnée en cache possède une durée de vie et des règles d'expiration.
- Il existe un mécanisme de rafraîchissement du cache.
- La stratégie de gestion de cache est inscrite dans la déclaration d'écoconception du service.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=102
- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=20

SP-API-05 – Bien concevoir et documenter pour augmenter le taux de réutilisation

Sprint Planning / API / Maintien et Évolution
Complexité : 3 – Priorité : 2

Le but est d'augmenter le taux de réutilisation de l'API et d'améliorer le temps de mise sur le marché des services qui en dépendent. Une documentation précise et détaillée permet aux utilisateurs de comprendre rapidement comment interagir avec l'API, ce qui facilite son adoption et son intégration.

Moyens de mise en œuvre :

- Créez une documentation exhaustive qui couvre tous les aspects de vos API, y compris :

1. Des exemples d'utilisation
2. Une sandbox de test
3. Des guides de démarrage rapide
4. Des FAQ

Critères de validation :

- La documentation de l'API couvre la totalité des cas d'usage
- La documentation de l'API est mise à jour à chaque changement du contrat d'interface
- La documentation de l'API comporte des exemples parlants
- La documentation de l'API comporte une sandbox pour expérimenter rapidement celle-ci

Liens :

- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=32

Sprint Planning / API / Maintien et Évolution

Complexité : 3 – Priorité : 3

Il arrive que certaines API d'un système d'information ne soient peu ou plus utilisées à la suite d'une suppression de fonctionnalité ou un arrêt d'application, mais qu'elles restent actives. Cela entraîne une sollicitation inutile de ressources informatiques pour des composants logiciels qui ne servent plus à rien. Il est donc essentiel de considérer la phase de décommissionnement comme une étape à part entière du cycle de vie des API, et ce dès la phase de conception.

Moyens de mise en œuvre :

- Identifier les API qui n'ont pas été utilisées depuis un certain temps en analysant les journaux d'accès ou les statistiques d'utilisation.
- Analyser les causes de non-utilisation afin de déterminer si des améliorations pourraient leur redonner une seconde vie.
- Évaluer l'impact de la désactivation de chaque API sur les systèmes et services qui pourraient en dépendre.
- Planifier le décommissionnement en informant les parties prenantes.
- Procéder à la désactivation des API identifiées, en s'assurant que toutes les dépendances sont prises en compte et que les utilisateurs sont informés.

Critères de validation :

- Aucune API non utilisée n'est en fonctionnement.
- Les ressources informatiques allouées aux API décommissionnées ont été libérées et leur utilisation a été vérifiée.

Liens :

- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=28

Sprint Planning / Architecture / Optimisation des performances

Complexité : 3 – Priorité : 3

L'objectif ici est de s'assurer que seules les données réellement nécessaires sont conservées. Les données dormantes occupent de l'espace disque inutile et alourdissent les traitements, il est donc important de procéder à un nettoyage régulier des anciennes données, que ce soit en les archivant pour une consultation future ou en les supprimant complètement si elles ne sont plus pertinentes. La manière dont les données sont utilisées et leur importance relative influencent les choix de type de stockage qui ne sont pas tous équivalents d'un point de vue de l'empreinte (un SDD est plus impactant en émission de gaz à effet de serre qu'un disque mécanique).

Moyens de mise en œuvre :

- Établir des processus d'archivage clairs en définissant une durée de vie pour chaque donnée.
- Mettre en place des mécanismes pour éviter les doublons dans les tables de données.

Critères de validation :

- Les processus d'archivage sont bien définis et appliqués de manière régulière.
- Les sauvegardes sont testées régulièrement pour s'assurer qu'elles peuvent être restaurées sans problème.
- La taille des espaces de stockage est surveillée et gérée de manière proactive.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 23

Sprint Planning / Architecture / Optimisation des performances

Complexité : 3 – Priorité : 3

HTML5 introduit une nouvelle fonctionnalité appelée ServiceWorker. Il s'agit d'un système de cache sous forme d'un proxy entre l'application web et le réseau. Grâce à lui, on peut intercepter les requêtes HTTP avant qu'elles ne soient envoyés au réseau et créer un système de cache intelligent grâce aux technologies de localstorage, sessionstorage, indexedDB. Cela permet de réduire considérablement la quantité de données échangées après le chargement initial, car on évite de retélécharger les éléments qui ne changent pas ou très peu.

En 2020, Philip Walton a partagé sur son blog des résultats impressionnants concernant cette approche. Il a observé une réduction des échanges réseau d'environ 48 %. De plus, il a noté une amélioration de plus de 52 % des performances, mesurée par le temps nécessaire pour afficher le premier contenu significatif (First Contentful Paint).

Moyens de mise en œuvre :

Tous les navigateurs récents supportent par défaut les ServiceWorker. Pour plus d'information sur la manière de les intégrer : https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API

Critères de validation :

- Le service utilise Le ServiceWorker.
- Les données qui ne nécessitent pas d'être mise à jour régulièrement sont chargées via Le ServiceWorker.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 59

SP-AR-03 – Utiliser une architecture événementielle pour éviter le "polling madness"

Sprint Planning / Architecture / Optimisation des performances
Complexité : 3 – Priorité : 3

Le polling madness est un pattern de conception consistant à appeler un endpoint à intervalle régulier afin d'actualiser des données, et potentiellement déclencher un évènement. Cela engendre une charge de travail supplémentaire et pousse souvent à augmenter les ressources informatiques pour gérer cette surcharge, d'autant qu'il est estimé que 98.5% des requêtes pull sont des requêtes gâchées. En adoptant une architecture événementielle, on inverse ce fonctionnement en demandant au système détenteur de l'information d'informer les services en ayant besoin de sa mise à jour. Ceux-ci reçoivent une notification dès qu'une information est modifiée/accessible.

Moyens de mise en œuvre :

Identifiez les événements clés qui déclenchent des modifications de données dans votre application, puis choisir une technologie adaptée pour gérer les événements, comme des systèmes de messaging (par exemple, Kafka, RabbitMQ) qui permettent de diffuser les notifications. Une fois la technologie choisie, intégrez des mécanismes d'écoute dans votre application pour capter ces événements et réagir en conséquence, en mettant à jour les données uniquement lorsque cela est nécessaire.

Critères de validation :

- Les événements applicatifs sont clairement identifiés.
- Les notifications d'événements sont correctement mises en place.
- L'application ne fait plus de requêtes régulières pour des données qui peuvent être obtenues via des événements.

Liens :

- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=14

SP-BE-01 – Mettre en cache les données calculées souvent utilisées

Sprint Planning / Backend / Optimisation des performances

Complexité : 3 – Priorité : 3

Lorsque nous traitons de gros volumes de données ou lançons des calculs qui demandent beaucoup de ressources en backend, il devient très vite avantageux de mettre en place un cache sur celles-ci, surtout si ces valeurs ne changent pas fréquemment. En utilisant un système de cache, comme un key-value store, on peut stocker ces résultats en mémoire vive (RAM), ce qui permet d'éviter de répéter des opérations coûteuses en termes de cycles CPU.

Moyens de mise en œuvre :

- Identifier les données backend avec une faible fréquence de rafraichissement ou demandant une forte sollicitation de ressources.
- Définir la stratégie de mise à jour.
- Mettre en place ou utiliser un mécanisme de cache.

Critères de validation :

- Les données peu volatiles, nécessitant un calcul et accédées plusieurs fois, sont placées dans un système de cache.
- Les temps de réponse des requêtes pour les données mises en cache sont significativement réduits par rapport aux requêtes qui nécessitent un recalcul.
- Les cycles CPU utilisés pour traiter les requêtes avec des données mises en cache sont nettement inférieurs à ceux utilisés pour les requêtes sans cache.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 16

SP-BE-02 – Autoriser un rafraîchissement partiel du cache et l'aligner sur le rafraîchissement des données

Sprint Planning / Backend / Optimisation des performances
Complexité : 2 – Priorité : 3

Lorsqu'on configure un cache, il faut veiller à ce que la politique de rafraîchissement des données (TTL, ou Time To Live) soit en phase avec le cycle de vie des données elles-mêmes. Dans le cas contraire, cela peut entraîner une inefficacité du cache car les données sources peuvent soit expirer trop tôt, ce qui signifie que l'on risque de travailler avec des informations du cache qui sont devenues obsolètes, soit expirer trop tard, ce qui peut engendrer un rechargement du cache inutile.

En permettant un rafraîchissement partiel des données mises en cache avec un cycle de rafraîchissement adapté, on peut maximiser l'efficacité du système de cache.

Moyens de mise en œuvre :

1. Analyser le cycle de vie des données pour déterminer les moments précis où les données doivent être rafraîchies.
2. Configurer le cache avec une politique de TTL qui reflète ces cycles de vie, en évitant les valeurs par défaut qui ne tiennent pas compte de la spécificité des données.
3. Mettre en place un mécanisme de rafraîchissement partiel, permettant de mettre à jour uniquement les données qui ont changé, plutôt que de vider et de recharger l'intégralité du cache.
4. Intégrer un système d'événements qui permet à la source de données d'informer le cache des changements, afin que celui-ci puisse agir en conséquence et maintenir des données à jour sans surcharge.

Critères de validation :

- Le cache est configuré avec une politique de TTL qui correspond au cycle de vie des données.
- Des mécanismes de rafraîchissement partiel sont en place.

Liens :

- https://www.collectif-api-thinking.com/assets/deliverables/worksites/50_CAT_API_Sustainable_IT.pdf#page=22

SP-BE-03 – Éviter le transfert de grandes quantités de données

Sprint Planning / Backend / Optimisation des performances

Complexité : 3 – Priorité : 3

Lorsque des traitements impliquent une logique complexe, il est préférable de les effectuer le plus près possible de la source de données pour réduire au maximum la sollicitation du réseau. Cela permet généralement de tirer parti des optimisations offertes par le SGBD pour la manipulation des données, ce qui contribue à alléger la charge CPU sur les serveurs, qu'ils soient backend ou frontend.

Moyens de mise en œuvre :

- Utiliser les procédures stockées pour les requêtes complexes qui manipulent un volume important de données.
- Privilégier d'effectuer les filtrages et les parcours de résultats de requêtes via le SGBD plutôt que de manière algorithmique.

Critères de validation :

- Les traitements complexes sont effectués au plus près de la source de données.
- Les procédures stockées sont utilisées pour les requêtes complexes.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 17

SP-BE-04 – Mutualiser les connexions à la base de données

Sprint Planning / Backend / Optimisation des performances

Complexité : 3 – Priorité : 3

L'ouverture d'une connexion à une base de données, quel que soit le type de système utilisé, représente une opération qui consomme des ressources, tant pour le client que pour le serveur. Ce processus implique plusieurs étapes : il faut allouer de la mémoire, gérer l'accès au disque pour les buffers, et effectuer des échanges de données sur le réseau pour établir le protocole de connexion. Pour optimiser cette gestion des connexions et améliorer les performances, il est préférable d'utiliser un pool de connexions afin de réutiliser des connexions existantes plutôt que d'en ouvrir de nouvelles à chaque fois.

Moyens de mise en œuvre :

Pour en savoir plus sur le Connection Pooling : <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/sql-server-connection-pooling>

Critères de validation :

- Les connexions sont réutilisées sans ouverture inutile de nouvelles connexions.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 24

SP-BE-05 – Optimiser les requêtes aux bases de données

Sprint Planning / Backend / Optimisation des performances

Complexité : 3 – Priorité : 3

L'optimisation des requêtes joue un rôle dans la limitation de la sollicitation des ressources des serveurs de base de données. Par exemple, il est toujours judicieux de limiter la quantité de données récupérées avant une agrégation en ne filtrant que ce qui est strictement nécessaire à celle-ci. L'ajout d'index sur les champs utilisés comme clés peut considérablement diminuer la puissance de calcul nécessaire pour exécuter une requête en plus d'accélérer son exécution.

Moyens de mise en œuvre :

Une analyse des requêtes par le biais d'outils comme EXPLAIN pour les systèmes de gestion de bases de données relationnelles (SGBDR), permet d'identifier les points d'amélioration facilement actionnables, bien que ne pouvant se substituer à une revue par un expert.

Critères de validation :

- Les temps d'exécution des requêtes et les ressources système (CPU, mémoire) utilisées lors de l'exécution des requêtes sont monitorés et restent sous un certain seuil d'acceptabilité.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 65

SP-BE-06 – Proposer un traitement asynchrone lorsque c'est possible

Sprint Planning / Backend / Optimisation des performances

Complexité : 3 – Priorité : 3

Lorsqu'une interaction avec l'utilisateur nécessite un traitement lourd et prolongé sur le serveur, il peut être intéressant d'opter pour un traitement asynchrone. L'utilisateur peut alors initier le processus sans avoir à rester devant son écran dans l'attente de la fin de celui-ci. Par exemple, après avoir lancé une tâche, l'utilisateur pourrait recevoir un courriel l'informant que le traitement est achevé, avec un lien pour accéder aux résultats.

Cette méthode permet d'effectuer des traitements par lots, qui peuvent être plus efficaces en termes de ressources que des traitements en temps réel. En libérant les serveurs de présentation, on permet à ces derniers de servir d'autres utilisateurs pendant que le traitement se fait en arrière-plan. Cela aide également à répartir la charge sur le serveur responsable du traitement, ce qui peut réduire le besoin en serveurs supplémentaires à quantité de travail équivalent.

Moyens de mise en œuvre :

- Identifier les traitements lourds qui pourraient être effectués de manière asynchrone.
- Valider avec le métier si l'interruption du parcours utilisateur est une stratégie acceptable pour le produit.
- Séparer le système de traitement des autres systèmes sur un serveur dédié.
- Mettre en place le mécanisme de notification.
- Permettre la reprise du parcours utilisateur via la notification.

Critères de validation :

- Le service ne réalise pas de traitements synchrones qui prennent plus d'une minute.
- Les traitements asynchrones sont mis en œuvre pour toutes les tâches lourdes identifiées.
- Les utilisateurs reçoivent une notification une fois le traitement asynchrone est terminé.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 7

SP-BE-07 – Prendre en compte les contraintes externes pour minimiser l’impact des calculs et transferts de données asynchrones

Sprint Planning / Backend / Optimisation des performances

Complexité : 3 – Priorité : 2

En planifiant intelligemment les calculs et les transferts de données asynchrones, comme les sauvegardes, les mises à jour ou l’entraînement d’algorithmes, afin de les réaliser en dehors des périodes de forte consommation électrique, on peut réduire la tension sur le réseau électrique, et possiblement l’intensité carbone du mixe électrique de la région du datacenter. En ajustant le timing de ces opérations, on peut aussi éviter les moments où les serveurs et les réseaux sont le plus sollicités, ce qui contribue à éviter l’achat de nouveaux équipements pour gérer les pics de demande. Enfin, il est préférable de privilégier les transferts de données lorsque les terminaux mobiles sont connectés à un réseau wifi plutôt que mobile.

Moyens de mise en œuvre :

Configurer les calculs et les transferts de données asynchrones en tenant compte de paramètres comme :

- L’Énergie : Éviter de programmer les calculs ou les transferts de données asynchrones pendant les pics de consommation électrique, moments où la production d’électricité est souvent plus carbonée. L’API Ecowatt, développée par RTE et l’ADEME, fournit des informations sur la disponibilité de l’énergie.
- L’Infrastructure : Décaler les calculs et les mises à jour importantes vers des périodes où la demande sur les ressources de calcul est plus faible.
- Le Réseau (heure) : Limiter les transferts de données importants entre 19h00 et 23h00, périodes généralement plus chargées. Par exemple, si une mise à jour volumineuse est prête à être déployée à 18h00, il serait préférable de la programmer pour 23h00.
- Le Réseau (type) : Lorsqu’un terminal mobile a la possibilité de se connecter à la fois à un réseau mobile et à un réseau wifi, il est recommandé de reporter les transferts de données non urgents (comme les mises à jour non critiques ou les sauvegardes) jusqu’à ce que le terminal soit connecté au réseau wifi.

Critères de validation :

- Le service numérique présente une déclaration d’écoconception qui explique clairement comment Les méthodes de décalage des calculs et des transferts de données asynchrones ont été mises en œuvre.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=120

SP-BE-08 – Éviter d'effectuer des requêtes SQL à l'intérieur d'une boucle

Sprint Planning / Backend / Gestion des données

Complexité : 3 – Priorité : 3

Les requêtes SQL exécutées à l'intérieur d'une boucle entraînent des problèmes de performance significatifs. Les serveurs SQL sont bien conçus pour gérer efficacement plusieurs opérations de sélection, d'insertion ou de modification en une seule requête ou transaction.

Moyens de mise en œuvre :

Regrouper les données à insérer ou à modifier dans une seule requête avant de les envoyer au serveur SQL. Par exemple, au lieu d'écrire un code qui exécute une requête pour chaque utilisateur dans une liste, il est préférable de construire une seule requête qui insère tous les utilisateurs en une seule opération.

Au lieu de faire ceci:

```
```PHP
foreach ($userList as $user) {
 $query = 'INSERT INTO users (first_name,last_name) VALUES(' . $user['first_name'] . ', ' . $user['last_name'] . ')';
 mysql_query($query);
}
```
```

Vous pouvez faire cela:

```
```PHP
$userData = array();
foreach ($userList as $user) {
 $userData[] = '(' . $user['first_name'] . ', ' . $user['last_name'] . ')';
}
$query = 'INSERT INTO users (first_name,last_name) VALUES ' . implode(',', $userData);
mysql_query($query);
```
```

Critères de validation :

- Le service n'exécute pas de requêtes SQL à l'intérieur d'une boucle.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 64

Sprint Planning / Backend / Gestion des données

Complexité : 3 – Priorité : 3

Le choix du type de données impacte la mémoire et le stockage. Utiliser un type de données trop large pour des valeurs qui pourraient être stockées dans un format plus compact peut nuire aux performances, que ce soit lors des opérations sur la base de données, au niveau du serveur d'applications ou même dans le navigateur. Il faut baser le choix du type de données et de son dimensionnement sur une analyse d'un échantillon représentatif des données à traiter.

Moyens de mise en œuvre :

Mener une analyse statistique pour évaluer les besoins réels en termes de capacité des champs de stockage de données. Dans une application en production depuis longtemps, il sera relativement facile de comparer la capacité des champs de données avec leur contenu réel en vue d'un redimensionnement.

Par exemple pour stocker un nombre de participant à une réunion, il pourrait être judicieux d'opter pour un type de données TINYINT, qui ne nécessite qu'un octet et peut contenir des valeurs jusqu'à 127, plutôt que de choisir un SMALLINT, qui utilise deux octets et peut aller jusqu'à 32 767. Le choix par défaut d'un INT, qui utilise quatre octets et peut stocker des valeurs jusqu'à 2 147 483 647, ou même d'un BIGINT, qui utilise huit octets, est souvent une erreur que l'on constate fréquemment lors des audits.

De même, il est inutile de stocker un courriel dans un type TEXT, un VARCHAR (50) suffit amplement.

Critères de validation :

- Les champs de la base de données ont tous un format adapté à leur contenu.
- Une analyse statistique a été réalisée pour justifier le choix des types de données utilisés.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 62

SP-BE-10 – Définir une politique d'expiration et suppression des données

Sprint Planning / Backend / Gestion des données

Complexité : 2 – Priorité : 2

Depuis l'entrée en vigueur du RGPD, il est devenu obligatoire de supprimer les données personnelles des utilisateurs après une certaine période. En plus de respecter la législation, cette pratique permet de réduire la sollicitation de ressources en réduisant les volumes de stockage des données. De nombreux services accumulent des quantités énormes de données qui ne sont jamais consultées, soit parce que l'utilisateur a cessé d'utiliser le service sans supprimer son compte, soit parce que les données sont devenues obsolètes.

Moyens de mise en œuvre :

- Définir des dates d'expiration sur chaque données (fichiers, entrées en base de données, etc.) pour permettre leur archivage ou suppression une fois la durée de conservation dépassée.
- Mettre en place un processus d'archivage ou de suppression des données dont la durée de conservation est dépassée.
- Déplacer, si nécessaire, les archives (données « froides ») sur un autre support que celui utilisé pour les données actives (données « chaudes »), par exemple des stockages coldline.

Critères de validation :

- Des dates d'expiration sont définies pour toutes les données stockées.
- Un mécanisme d'archivage ou de suppression des données dépassant la durée de conservation définie est mis en place.
- Un suivi de l'évolution du poids des fichiers stockés et des bases de données est effectué régulièrement.
- Le service possède deux types de systèmes de stockage de données : chaudes et froides.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 78
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=103

Sprint Planning / Contenus / Optimisation des médias

Complexité : 2 – Priorité : 2

Une fois qu'ils sont dépubliés, conserver contenus multimédias n'a généralement plus d'intérêt. C'est pourquoi il est utile de mettre en place une politique de suppression des contenus pour éviter de surcharger inutilement le stockage et libérer de l'espace pour de nouveaux. En allégeant les bases de données et les serveurs physiques des contenus non utiles, on évite de devoir redimensionner vers le haut les infrastructures au long de la vie du service.

Moyens de mise en œuvre :

- Supprimer les contenus dépubliés des supports de stockage actifs de vos serveurs. Si ces contenus doivent être conservés, utilisez des supports de stockage froids
- Définir une stratégie d'archivage et de suppression des contenus obsolètes, périmés, dépassés ou inutiles à conserver en définissant une date d'expiration et un processus d'archivage et/ou de purge automatique.

Critères de validation :

- Il n'existe pas de contenus stockés qui ne sont pas accessibles sur les pages web.
- Une stratégie d'archivage et de suppression est clairement définie.
- Des mécanismes automatiques et des processus manuels pour les contenus nécessitant une intervention humaine sont en place.
- Le taux d'occupation des bases de données et des serveurs physiques est régulièrement évalué pour suivre l'efficacité de la stratégie mise en place.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 114
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=91

Sprint Planning / Contenus / Optimisation des médias

Complexité : 2 – Priorité : 2

On souhaite réduire la quantité de données téléchargées et la consommation d'énergie des terminaux en optimisant et dimensionnant les images, les audios et les vidéos. Il s'agit de trouver un équilibre entre une expérience utilisateur de qualité et une minimisation des ressources sollicitées par le service.

Moyens de mise en œuvre :

- Pour les images : Les images matricielles doivent être affichées dans leur taille d'origine ou proposées en plusieurs définitions pour s'adapter au contexte d'affichage. Utilisez les attributs HTML ``srcset`` et ``sizes`` pour permettre au navigateur de choisir la meilleure image en fonction de la résolution de l'écran.
- Pour l'audio : Utilisez des codecs audios optimisés comme Opus, AAC ou Dolby AC-4. Optimisez le bitrate, le ratio de compression et la fréquence d'échantillonnage. Adaptez le nombre de canaux en fonction du contenu : stéréo pour la musique, mono pour les dialogues. Évitez un ratio poids en mégaoctet/durée en minute supérieur à 1.
- Pour la vidéo : Proposez des vidéos avec des codecs performants comme webm (AV1, VP9) ou HEVC, et utilisez un encodage à débit variable. Adaptez la définition vidéo au contexte de visualisation, avec une définition par défaut ne dépassant pas le mode "qualité standard". Proposez un mode "sobriété énergétique" facilement accessible, voir en option par défaut, qui réduit la définition pour économiser des données et de l'énergie.

Critères de validation :

- Les images du service numérique sont soit vectorielles, soit matricielles affichées dans leur taille d'origine, soit proposées en plusieurs variantes avec des définitions différentes adaptées au contexte d'affichage.
- Les contenus audios sont proposés avec des codecs optimisés, et les détails de l'encodage audio (débit, nombre de canaux) sont documentés dans la déclaration d'écoconception du service numérique.
- Les vidéos sont encodées avec un débit variable et proposées avec des codecs performants pour les définitions 720p et plus. Les détails de l'encodage vidéo et audio sont documentés dans la déclaration d'écoconception.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 47
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=29>
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=33>
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=82
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=84
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=89
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=97

SP-CO-03 – Forcer ou encourager l'optimisation des médias avant importation sur une application web ou un CMS

Sprint Planning / Contenus / Optimisation des médias

Complexité : 1 – Priorité : 3

L'idée ici est de s'assurer que tous les médias (images, vidéos, audios, documents) sont optimisés avant même de les importer dans le CMS, car les plugins et extensions dédiés des Framework CMS peuvent être gourmands en ressources (CPU, mémoire vive, réseau, stockage) et ne garantissent pas toujours une optimisation parfaite. En optimisant les fichiers en amont, on s'assure que le système en production ne pourra pas proposer de médias disproportionnés ou inadaptés qui encombrerait le stockage et le réseau.

Moyens de mise en œuvre :

- Cela peut se faire en contraignant le format et la taille des fichiers dans les champs d'import du backoffice, et en rappelant à l'administrateur l'importance de cette pratique (on peut même lui fournir un lien vers un outil de compression adapté dans la description du champs d'upload).
- Pour les PDF, assurez-vous que les images sont correctement compressées et à une résolution maximale de 72 dpi. N'incluez que les principales polices pour alléger le fichier. Si le document est encore trop volumineux, découpez-le en chapitres pour limiter les téléchargements inutiles.

Critères de validation :

- Tous les contenus sont optimisés avant d'être importés dans le Framework ou le CMS.
- L'interface d'administration rappelle cette bonne pratique partout où l'on peut importer du contenu.
- L'interface d'administration force un certain niveau d'optimisation, en limitant les formats et la taille des fichiers à l'import

Liens :

- Les 115 bonnes pratiques d'écoconception web : 97
- Les 115 bonnes pratiques d'écoconception web : 109

Sprint Planning / Contenus / Optimisation des médias

Complexité : 3 – Priorité : 3

Choisir le bon format d'image évite une charge inutile sur le réseau et accélère le chargement des pages. Les images vectorielles, composées de formes géométriques, peuvent être agrandies à l'infini sans jamais être pixellisées. Elles pèsent moins lourd que les photos ou images matricielles.

L'on peut également remplacer les images matricielles (GIF, PNG, JPEG, WebP, etc.) par des styles (CSS), des pictos, des glyphes ou des icônes fournis par une webfont ou une police standard. Le client web n'a ainsi aucune ressource supplémentaire à télécharger.

Moyens de mise en œuvre :

- Choisir le format adapté :
 - Format vectoriel (exemple SVG) : Utiliser ce format lorsque cela est possible (illustrations, icônes, logos, graphes...).
 - Format matriciel, compression sans perte (lossless) : Pour des illustrations avec aplats de couleur, remplacer le PNG par des images WebP ou JPEG XL en mode « compression sans perte ».

• Format matriciel, compression avec perte (lossy) : Pour des photos ou des illustrations avec plusieurs milliers de couleurs, remplacer le JPEG par des images WebP, AVIF ou JPEG XL en mode « compression avec perte ».

Il est possible de déterminer en quelques secondes le potentiel de réduction du poids total des images grâce à des changements de formats grâce à des services d'audit d'images tel que <https://websitespeedtest.cloudinary.com/>

- Minifier et optimiser les fichiers .svg en utilisant des outils comme compressor.io. Si un fichier .svg est très complexe, il peut être plus lourd qu'un fichier .png.
- Utiliser des glyphes quand c'est possible, ou des icônes et styles CSS pour réduire la bande passante, le nombre de requêtes et la complexité du DOM.
- Utiliser les iconfont : Un gain de 91 % du poids a été réalisé sur une page web en utilisant exclusivement des pictogrammes et illustrations vectorielles dans une icon font via FontForge (<https://fontforge.org/>).

Critères de validation :

- Le service n'utilise pas plus que 5 images matricielles par pages.
- Plus de 75 % des images matricielles utilisées pour le service numérique sont dans un format efficace (WebP, AVIF, JPEG XL ou un format d'image plus performant).
- Toute image qui auraient pu être remplacées par un glyphe (symbole de police de caractère) a été remplacée.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 48
- Les 115 bonnes pratiques d'écoconception web : 49
- Les 115 bonnes pratiques d'écoconception web : 100
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=79
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=90
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=30>

SP-CO-05 – Compresser les images

Sprint Planning / Contenus / Optimisation des médias

Complexité : 3 – Priorité : 3

La compression d'image consiste à réduire la quantité de données qui composent l'image, tout en préservant une qualité visuelle acceptable pour l'œil humain. Deux nuances de noir très proches peuvent être unifiées par un outil de compression en une seule couleur, rendant la différence imperceptible en diminuant le poids de l'image. Il existe différents types de compression : la compression "lossy", qui offre un gain de poids maximal avec une légère perte de qualité, et la compression "lossless", qui préserve la qualité de l'image tout en réduisant son poids de manière moins significative. Le choix entre ces deux méthodes dépendra de l'usage prévu de l'image ; par exemple, pour des images contenant du texte ou des détails importants, la compression "lossless" est préférable, tandis que pour des photos, la compression "lossy" peut suffire.

Moyens de mise en œuvre :

Plusieurs outils en ligne existent pour compresser les images, tels que CloudConvert, ImageCompressor ou TinyPNG. Dans un système de gestion de contenu (CMS) comme WordPress, des plugins tels que shortpixel.com ou imagify.io intègrent le processus de compression directement depuis l'interface de gestion.

Gardez à l'esprit que certains effets visuels peuvent nuire à la compréhension des images pour des personnes ayant des handicaps visuels.

Critères de validation :

- Les images sont compressées avec un outil approprié, en choisissant le type de compression adapté à leur contenu.
- Le poids total des images est réduit de manière non négligeable par rapport à leur taille originale.
- La qualité visuelle des images est maintenue à un niveau acceptable, sans perte significative de détails importants.
- Les images sont accessibles et compréhensibles pour tous les utilisateurs, y compris ceux rencontrant des handicaps visuels.

Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=31>

Sprint Planning / Contenus / Optimisation des médias

Complexité : 3 – Priorité : 3

Les images vectorielles au format SVG (Scalable Vector Graphics) sont basées sur des vecteurs décrits en XML et peuvent être redimensionnées à l'infini sans perdre en qualité, solution parfaite pour les designs responsives. Cependant, les fichiers SVG peuvent contenir des métadonnées superflues, qui, bien qu'elles soient utiles lors de la création et édition, alourdissent le fichier final. En supprimant ces métadonnées, on peut significativement réduire le poids du fichier à transférer. Il est possible d'obtenir une réduction de taille de fichier allant jusqu'à 75 % dans certains cas.

Moyens de mise en œuvre :

Lors d'un test avec SVGO sur un fichier SVG de 1 Ko, le poids de celui-ci a été réduite de 36 %, passant de 1 101 à 700 octets. En appliquant ensuite une compression Gzip avant le transfert, le poids du fichier chute à 498 octets, soit moins de la moitié de la taille initiale, tout cela sans aucune perte de qualité à l'affichage.

Critères de validation :

- Tous les fichiers SVG ont été traités par un outil d'optimisation
- Les fichiers SVG optimisés sont compressés avec Gzip avant leur transfert

Liens :

- Les 115 bonnes pratiques d'écoconception web : 99

Sprint Planning / Contenus / Optimisation des médias

Complexité : 3 – Priorité : 3

Une part importante du trafic vidéo sur Internet est générée par des contenus qui sont écoutés plutôt que regardés. Des études, comme celle menée par Chris Priest de l'université de Bristol en mai 2019, montrent que de nombreuses vidéos, telles que des clips musicaux ou des podcasts filmés, sont souvent utilisées en fond sonore. En offrant aux utilisateurs la possibilité de désactiver le flux vidéo, en optant pour un mode « écoute seule », « audio only » ou « zéro pixel », on diminue drastiquement le volume de données transférées, mais aussi la consommation d'énergie des équipements utilisateur, en permettant l'écoute de l'audio même lorsque l'écran est éteint.

Moyens de mise en œuvre :

Proposer une option sur le lecteur vidéo, permettant à l'utilisateur de passer facilement en mode « écoute seule » et de revenir au mode vidéo quand il le souhaite. En mode « écoute seule », le lecteur vidéo doit cesser de télécharger le flux vidéo, se concentrant uniquement sur le flux l'audio. Il peut être intéressant d'automatiser cette transition, par exemple si l'utilisateur change d'onglet dans un navigateur ou éteint l'écran de son appareil.

Critères de validation :

- Le service propose un mode « écoute seule » sur les vidéos, qui peut être activé par l'utilisateur et qui s'enclenche automatiquement lorsque la vidéo n'est pas affichée sur le terminal (par exemple, changement d'onglet, passage à une autre application, ou extinction de l'écran).
- Dans le cas où les vidéos sont hébergées par un service tiers ne proposant pas ce mode, il est possible de valider le critère en fournissant un enregistrement « écoute seule » sous chaque vidéo, sans précharger la vidéo associée.

Liens :

- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=87

SP-FE-01 – Avoir un titre de page et une metadescription pertinents

Sprint Planning / Frontend / Structure et Contenu

Complexité : 3 – Priorité : 3

Le titre de page <h1>, ainsi que son équivalent <title>, adjoints à une balise <meta name="description"> pertinente, doivent être parfaitement en accord avec le contenu de la page associée.

Cette optimisation répond à l'intention de recherche de l'utilisateur, lui évitant une visite inutile (des allers-retours pour trouver l'information recherchée). Elle raccourcit aussi son temps de parcours au sein des moteurs de recherche.

Soigner la qualité des éléments <title> et <meta name="description"> permet également :

- d'indexer correctement les pages de contenus par les moteurs de recherche et les annuaires;*
- d'assurer l'identification du site et du contenu de la page d'atterrissage;*
- de garantir la compréhension avant l'accès au contenu;*
- de qualifier la lecture du contenu ou de l'information demandée au sein de la page associée.*

Moyens de mise en œuvre :

Assurer la pertinence des balises page <h1>, <title>, <meta name="description"> en accord avec le contenu de la page associée.

Critères de validation :

- Les titres de pages sont tous repris dans la balise TITLE.
- Les balises <title> et <meta name="description"> sont pertinentes et en adéquation avec le contenu de la page.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 11

SP-FE-02 – Favoriser les pages statiques

Sprint Planning / Frontend / Structure et Contenu

Complexité : 3 – Priorité : 2

L'idée est de privilégier l'utilisation de pages statiques pour les contenus qui ne nécessitent que peu de mises à jour, par exemple, celles qui ne changent que deux fois par an. En optant pour des pages statiques, qui sont créées en dehors d'un système de gestion de contenu (CMS), on parvient à réduire la sollicitation de ressources, notamment en termes de cycles CPU et de bande passante. En effet, un CMS dynamique nécessite de charger plusieurs couches logicielles pour afficher le contenu demandé par l'utilisateur. À l'inverse, une page statique est simplement lue et renvoyée directement par le serveur HTTP ou le serveur de cache, sans avoir à solliciter le serveur d'applications ou la base de données.

Moyens de mise en œuvre :

En fonction du service numérique à concevoir, il est envisageable :

- pour une landing page ou simple site vitrine, de créer un site entièrement statique en HTML, CSS et JS.
- pour un blog avec peu d'activité, ou de petite taille, de créer un site statique généré via un JAMstack (Jekyll, Hugo, Gatsby, Eleventy, etc.) <https://jamstack.org/> et administré, si nécessaire, via un headless CMS ou un headless flat-file CMS (Strapi, Contentful, Flextype, etc.).
- pour un site ou une application web plus complexe de rendre les pages à faible taux de modifications (FAQ, About, mentions légales, etc.) statiques en utilisant des Framework avec Server Side Rendering (Next, Nuxt, Svelte, etc.).

Critères de validation :

- Les pages statiques sont correctement mises en place et accessibles sans nécessiter de chargement dynamique.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 18

Sprint Planning / Frontend / Structure et Contenu

Complexité : 3 – Priorité : 2

Le sitemap joue un rôle crucial dans l'indexation des pages et des contenus d'un site web par les moteurs de recherche dans le cas où le service est publiquement accessible sur internet. Avec un sitemap bien structuré et régulièrement mis à jour, on réduit le risque que les internautes tombent sur des pages sans intérêt ou sur des pages orphelines.

Moyens de mise en œuvre :

Voici un exemple de structure de sitemap :

```
``xml
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
<loc>http://mon-domaine.fr/</loc>
<lastmod>2024-08-28</lastmod>
<changefreq>daily</changefreq>
<priority>1</priority>
</url>
<url>
<loc>http://mon-domaine.fr/page-a.html</loc>
<lastmod>2024-07-01</lastmod>
<changefreq>monthly</changefreq>
<priority>0.8</priority>
</url>
</urlset>
``
```

Dans cet exemple, chaque balise ``<url>`` représente une page du site, avec des informations sur son emplacement (`<loc>`), sa dernière modification (`<lastmod>`), la fréquence à laquelle elle est mise à jour (`<changefreq>`) et son niveau de priorité (`<priority>`).

Critères de validation :

- Il n'y a pas de pages orphelines dans le sitemap.
- Le sitemap est mis à jour régulièrement pour refléter les modifications apportées aux pages du site.
- Les balises ``<lastmod>`` sont correctement renseignées pour chaque URL, indiquant la dernière date de modification.
- Les fréquences de changement et les priorités sont définies de manière appropriée pour chaque page, en fonction de leur contenu et de leur importance.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 89

SP-FE-04 – Afficher des pages d'erreur statiques

Sprint Planning / Frontend / Structure et Contenu

Complexité : 1 – Priorité : 1

Rendre les pages d'erreur, comme celles qui renvoient des codes 404 ou 500, aussi légères que possible. Quand un navigateur cherche à accéder à une ressource qui n'existe pas, comme une image ou un fichier CSS, ou lorsqu'une erreur se produit sur le serveur, la page d'erreur qui est renvoyée peut parfois être plus lourde que la ressource initialement demandée.

Moyens de mise en œuvre :

Il est conseillé d'éviter les pages d'erreur qui sont générées dynamiquement et personnalisées en fonction du contenu de l'URL. Au lieu de cela, opter pour une page statique, qui ne nécessite pas d'interrogation de la base de données ni de traitement dynamique.

Critères de validation :

- Les pages d'erreur statiques sont correctement configurées et accessibles.
- Aucune page d'erreur personnalisée ne nécessite d'interrogation de la base de données.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 20

Sprint Planning / Frontend / Structure et Contenu

Complexité : 2 – Priorité : 3

Une CSS print a pour but de simplifier le contenu à imprimer en ne reprenant pas la totalité des contenus affichés à l'écran. En proposant une feuille de styles dédiée à l'impression, on se concentre uniquement sur le contenu qui aura de la valeur dans une impression papier, typiquement le texte d'un article ou l'image principale d'une page de photographie et on rend totalement invisible les éléments de navigation et de design de la page web. Ainsi, on réduit le nombre de pages nécessaires lors de l'impression, donc moins de papier gaspillé et moins d'encre consommée.

Moyens de mise en œuvre :

Créer une feuille de styles qui va "nettoyer" la page telle qu'elle est affichée à l'écran. Voici quelques exemples de règles CSS à intégrer :

- Définir un fond clair et une police de caractères lisible pour le corps du texte, par exemple :

```
``body {  
background-color: #fff;  
font-family: Serif;  
font-size: 15pt;  
}  
``
```

- Supprimer les marges et les bordures inutiles pour le conteneur principal :

```
``#page {  
margin: 0;  
border: none;  
}  
``
```

- Cacher les éléments non pertinents pour l'impression, tels que le bandeau supérieur, le menu latéral, et le pied de page :

```
``#banner, #menuright, #footer {  
display: none;  
}  
``
```

Critères de validation :

- Toutes les pages présentant du contenu possèdent une CSS Print.
- La feuille de styles est correctement appliquée lors de l'impression, assurant que les éléments non désirés sont masqués.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 31

SP-FE-06 – N'utilisez que les portions indispensables des bibliothèques JS et CSS

Sprint Planning / Frontend / Optimisation des Ressources

Complexité : 3 – Priorité : 3

Les bibliothèques JavaScript comme jQuery et les Framework CSS tels que Bootstrap, Skeleton, Gumby ou Foundation sont des outils très pratiques pour créer des sites web rapidement. Cependant nous n'exploitons souvent qu'une infime partie de ces bibliothèques. Certaines d'entre elles ne sont pas conçues de manière modulaire, ce qui oblige le client à télécharger l'intégralité de la bibliothèque, même si seules quelques fonctionnalités sont réellement nécessaires. Il est donc conseillé de se passer de ces bibliothèques lorsque cela est possible, ou de n'en conserver que les éléments réellement utilisés.

Moyens de mise en œuvre :

Plusieurs Framework proposent des versions allégées de leurs bibliothèques, qui ne contiennent que les éléments nécessaires pour le site. Cela permet de diminuer le poids des fichiers que le navigateur doit télécharger et de réduire la consommation de ressources lors de l'exécution. L'utilisation d'un bundler, comme Webpack, permet de réaliser du "tree shaking", une technique qui consiste à éliminer le code superflu et donc à réduire la sollicitation de ressources.

Critères de validation :

- Les versions allégées des bibliothèques sont mises en œuvre lorsque cela est possible.
- Un outil de bundling est utilisé pour effectuer le tree shaking et éliminer le code non utilisé.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 15

SP-FE-07 – Charger uniquement ce qui est nécessaire pour l’affichage et recharger partiellement les zones de contenu

Sprint Planning / Frontend / Optimisation des Ressources

Complexité : 3 – Priorité : 3

Nous pouvons être tenté de précharger des ressources en prévision d’une action utilisateur qui pourrait ne jamais se produire, ce qui sollicite des ressources inutilement. L’idée est donc de ne charger que les ressources quand elles sont nécessaires. Si le service comporte des zones de contenu qui nécessitent un rafraîchissement fréquent, il est préférable de ne recharger que ces zones spécifiquement plutôt que toute la page. Cela permet de conserver une solution de cache pour la page principale et de ne recharger que les parties dynamiques.

Moyens de mise en œuvre :

- Évitez d’utiliser des balises comme `<link rel="prerender">`, ainsi que tous les `rel="preload"`, `rel="prefetch"`, `rel="preconnect"`, `rel="modulepreload"` et `rel="dns-prefetch"` pour des ressources qui ne seront pas utilisées sur la page courante ou qui ne seront utilisées que sous certaines conditions.
- Découpez le code JavaScript et CSS en portions (chunks) relatives à des actions spécifiques de l’utilisateur et ne chargez ces portions que lorsqu’elles sont nécessaires. Ça peut être particulièrement efficace pour les applications monopage (SPA) afin d’éviter le chargement complet de l’application en une fois.
- Utilisez des ServiceWorker, des Web Components ou des développements sur-mesure pour recharger uniquement les zones de contenu qui nécessitent un taux de rafraîchissement élevé. Vous pouvez également adopter des architectures comme les "component islands" pour hydrater ces zones dynamiques en JavaScript sans inclure ces parties dans l’appel HTTP de la page.

Critères de validation :

- Toutes les ressources chargées dans le front sont utilisées lors d’un parcours utilisateur donné.
- Toutes les zones de contenu pouvant être mises à jour partiellement utilisent cette méthode.
- La réutilisation de données n’ayant pas de taux de rafraîchissement critique n’entraîne pas de nouvel d’appel réseau à chaque utilisation du service.

Liens :

- Les 115 bonnes pratiques d’écoconception web : 34
- Les 115 bonnes pratiques d’écoconception web : 45

Sprint Planning / Frontend / Optimisation des Ressources

Complexité : 3 – Priorité : 3

Chaque fichier CSS chargé dans une page web entraîne une requête HTTP, donc un échange réseau. En optant pour une feuille de styles unique pour les éléments communs, peu importe le type d'affichage, et en créant des fichiers spécifiques pour chaque résolution ou media query, on simplifie la gestion des styles et on réduit le nombre de fichiers CSS nécessaires au strict minimum.

Moyens de mise en œuvre :

Cet exemple de code charge les fichiers CSS appropriés en fonction de la taille de l'écran ou du type d'affichage :

```
``html
<link rel='stylesheet' id='css-css' href='communs.css' type='text/css' media='all' />
<link rel='stylesheet' id='css-xs-css' href='petits-ecrans.css' type='text/css' media='(max-width: 959px)' />
<link rel='stylesheet' id='css-sm-css' href='tablettes.css' type='text/css' media='(min-width: 768px)' />
<link rel='stylesheet' id='css-mdlg-css' href='grands-ecrans.css' media='(min-width: 960px)' />
<link rel='stylesheet' id='css-print-css' href='print.css' type='text/css' media='print' />
``
```

Critères de validation :

- La page se limite à un maximum de 10 fichiers CSS.
- Les fichiers CSS sont intégrés directement dans le code HTML sans utiliser de directives @import.
- Les fichiers CSS sont chargés en fonction des media queries appropriées pour optimiser le rendu sur différents appareils ou affichages.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 35

Sprint Planning / Frontend / Optimisation des Ressources

Complexité : 3 – Priorité : 3

A l'opposé de la pratique " limiter le nombre de CSS". L'idée ici est de ne pas tout regrouper dans un seul gros fichier CSS, mais plutôt de créer plusieurs fichiers CSS adaptés aux différents modules, spécialement concernant la gestion des pages mobiles ou de fonctionnalités très peu utilisées. En procédant ainsi, vous ne chargez que les styles nécessaires en fonction du contexte d'utilisation et donc vous réduisez le poids des fichiers CSS lors du premier chargement.

Moyens de mise en œuvre :

Commencez par segmenter vos fichiers CSS selon la logique fonctionnelle de votre site. Par exemple, vous pourriez créer des fichiers distincts pour :

- le layout (la structure générale de la page) ;
- des modules spécifiques (groupés en fonction de leur fréquence d'utilisation).

Cette méthode est particulièrement efficace pour les sites riches en fonctionnalités, car elle permet d'éviter de charger des styles inutiles pour des modules qui ne sont pas utilisés sur certaines pages. Les fichiers CSS généraux, comme ceux pour le layout et le contenu, peuvent être regroupés en un seul fichier, tandis que les fichiers CSS complémentaires, comme ceux des modules spécifiques, seront chargés dynamiquement en fonction des besoins de la page ou des fonctionnalités activées.

Critères de validation :

- Tout le CSS n'est pas regroupé dans un seul et même fichier.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 36

SP-FE-10 – Minifier, Combiner et compresser les fichiers CSS, JavaScript, HTML et SVG

Sprint Planning / Frontend / Optimisation des Ressources

Complexité : 3 – Priorité : 3

L'optimisation des fichiers CSS, JavaScript, HTML et SVG est un levier important pour améliorer les performances de votre service. En combinant, compressant et minifiant ces fichiers, vous réduisez le nombre de requêtes HTTP, économisez de la bande passante et accélérez les temps de chargement. Cela se traduit par une meilleure expérience utilisateur et une utilisation plus efficace des ressources réseau. Utiliser des algorithmes de compression comme Brotli ou GZIP peut encore améliorer ces gains de performance.

Moyens de mise en œuvre :

- Minification : Utilisez des outils de minification pour réduire la taille des fichiers en supprimant les espaces inutiles, les commentaires, les sauts de ligne, etc. Voici quelques outils recommandés :

- CSS : cssnano, cssso, clean-css
- JavaScript : Terser, UglifyJS, Babel-minify
- HTML : htmlnano, HTMLMinifier
- SVG : SVGO, minify-xml

- Combinaison :

- Avec Webpack, le plugin `Webpack-merge-and-include-globally` facilite la fusion des fichiers CSS et JavaScript.
- Dans WordPress, utilisez des extensions gratuites comme Autooptimize ou Assets CleanUp pour combiner les fichiers.

- Compression :

- Pour utiliser Brotli dans Apache, ajoutez la configuration suivante :

```
``apache
# compress html, text, xml, css, javascript
AddOutputFilterByType  BROTLI_COMPRESS  text/html  text/plain  text/xml  text/css  text/javascript
application/javascript
# Or, compress certain file types in a specific directory:
<Directory ""/your-directory"">
AddOutputFilterByType BROTLI_COMPRESS text/html
</Directory>
``
```

Certains fournisseurs de Content Delivery Network (CDN) proposent également d'activer Brotli.

Critères de validation :

- Tous les fichiers CSS, JavaScript, HTML et SVG sont minifiés, combinés et compressés.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 75
 - Les 115 bonnes pratiques d'écoconception web : 76
 - Les 115 bonnes pratiques d'écoconception web : 77
- https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=95

SP-FE-11 – Préférer les CSS aux images

Sprint Planning / Frontend / Optimisation des Ressources

Complexité : 3 – Priorité : 3

On peut tirer parti des propriétés CSS3 pour remplacer les images dans les pages et ainsi réduire considérablement le poids des fichiers à charger. En outre, l'appel d'une feuille de styles ne génère qu'une seule requête HTTP, contre un grand nombre si l'on emploie beaucoup d'images (une requête HTTP pour chaque image).

Un développeur s'est amusé à développer tous les drapeaux du monde en css, prouvant la capacité de créer des motifs complexes avec ce langage : <https://pixelastic.github.io/css-flags/>

Moyens de mise en œuvre :

Remplacez les éléments visuels qui pourraient être réalisés par des styles CSS, par exemple pour créer des formes géométriques.

Si vous souhaitez arrondir les coins d'un cadre, utilisez la propriété `border-radius` en CSS3 au lieu de créer une image de cadre avec des coins arrondis.

```
```css
#cadre {
 border-radius: 10px;
}
```
```

Critères de validation :

- Aucune image n'est utilisée pour des éléments qui peuvent être réalisés avec des propriétés CSS.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 37

Sprint Planning / Frontend / Optimisation des Ressources

Complexité : 3 – Priorité : 3

La compartimentation CSS est une technique qui permet de déclarer qu'un élément et son contenu peuvent fonctionner de manière relativement autonome par rapport au reste de la structure de la page. Cela signifie que lorsque vous appliquez cette propriété, le navigateur peut se concentrer sur le rendu de cet élément spécifique sans avoir à recalculer l'ensemble du Document Object Model (DOM) de la page. En fonction du type d'indépendance que vous choisissez (qu'il s'agisse de la taille, de la mise en page, du style ou du rendu), le navigateur peut optimiser le processus de rendu en recalculant uniquement les éléments nécessaires pour les propriétés de cet élément isolé. De plus, les contenus des nœuds qui sont marqués comme indépendants ne seront pas rendus tant qu'ils ne sont pas visibles à l'écran.

Moyens de mise en œuvre :

Identifiez les parties de votre architecture DOM qui sont complexes et dont le contenu peut être modifié à différents moments.

Par exemple, vous pouvez appliquer la propriété `contain: content;` à un article spécifique pour le rendre indépendant :

```
```css
article {
 contain: content;
}
```
```

De même, si vous avez un widget externe dont la mise en page doit être isolée, vous pouvez utiliser `contain: layout;` :

```
```css
.widget-name {
 contain: layout;
}
```
```

Pour approfondir : <https://www.w3.org/TR/css-contain-2/>

Critères de validation :

- Les éléments qui peuvent être isolés sont marqués avec la propriété `contain`.

Liens :

- Les 115 bonnes pratiques d'écoconception web : 38

SP-FE-13 – Modifier plusieurs propriétés CSS en une seule fois

Sprint Planning / Frontend / Optimisation des Ressources

Complexité : 2 – Priorité : 3

Chaque fois qu'une propriété CSS est modifiée individuellement, cela peut entraîner des opérations lourdes telles que des repaint et reflow. Ces opérations peuvent solliciter inutilement les ressources de calcul du système et ralentir la page.

Il est recommandé de toujours utiliser les classes CSS pour regrouper les modifications de plusieurs propriétés en une opération et restreindre le nombre de repaint et reflow à un seul.

Moyens de mise en œuvre :

Définir des classes CSS spécifiques à chaque état visuel de vos composants.

Par exemple :

```
```html
<style>
.in-error {
color: red;
font-weight: bold;
}
</style>
<script>
$.el.bind('error', function () {
$.el.addClass('in-error');
});
$.el.bind('running', function () {
$.removeClass('in-error');
});
</script>
```

### **Critères de validation :**

- Toutes les modifications de déclarations CSS qui pourraient être regroupées en une seule opération sont regroupés en une classe spécifique.
- Aucune propriété CSS n'est modifiée individuellement sans passer par l'ajout ou la suppression de classes.

### **Liens :**

- Les 115 bonnes pratiques d'écoconception web : 39

*Sprint Planning / Frontend / Optimisation des Ressources*

*Complexité : 3 – Priorité : 3*

*Des sélecteurs complexes ou trop spécifiques peuvent ralentir le rendu des pages. En optant pour des sélecteurs basés sur des ID ou des classes, on permet au navigateur de trouver nos composants plus rapidement.*

**Moyens de mise en œuvre :**

Ne pas écrire des sélecteurs basés sur des structures d'éléments.

Par exemple, au lieu d'utiliser un sélecteur comme :

- ``treeitem[mailfolder="true"] > treerow > treecell {...}``

il est préférable d'opter pour un sélecteur plus simple et direct tel que :

- ``.treecell-mailfolder {...}``

De même, pour les en-têtes de tableau, au lieu de :

- ``treehead > treerow > treecell {...}``

il est recommandé d'utiliser :

- ``.treecell-header {...}``

**Critères de validation :**

- Tous les sélecteurs CSS présents dans le code favorisent l'utilisation d'ID ou de classes.

**Liens :**

- Les 115 bonnes pratiques d'écoconception web : 40

*Sprint Planning / Frontend / Optimisation des Ressources*

*Complexité : 3 – Priorité : 3*

*Les codes CSS et JavaScript ne doivent pas être directement intégrés dans le code HTML des pages, sauf pour des variables de configuration spécifiques aux objets JavaScript. En plaçant ces codes dans des fichiers séparés, le navigateur aura la possibilité de les stocker dans son cache local. Ainsi, lors des visites ultérieures, il peut les récupérer sans avoir à les télécharger de nouveau.*

**Moyens de mise en œuvre :**

Ne pas inclure de balises de style ou de script directement dans le code HTML. Par exemple, au lieu d'écrire :

```
```html
<style type="text/css" media="screen">
p { color: #333; margin: 2px 0 }
/* Toutes les déclarations CSS du site */
</style>
```
```

il est préférable d'utiliser une balise de lien pour référencer un fichier CSS externe, comme ceci :

```
```html
<link href="css/styles.css" rel="stylesheet">
```
```

**Critères de validation :**

- Aucune page n'intègre une quantité significative de code CSS ou JavaScript.

**Liens :**

- Les 115 bonnes pratiques d'écoconception web : 41

---

## SP-FE-18 – Limiter le poids et le nombre de requêtes par écran

---

### *Sprint Planning / Frontend / Optimisation des Ressources*

*Complexité : 3 – Priorité : 2*

*Le temps de chargement d'une page côté navigateur est directement lié au nombre de fichiers que le navigateur doit télécharger et au poids de chaque fichier. Chaque fichier nécessite une requête GET HTTP vers le serveur, et plus il y a de requêtes, plus le serveur est sollicité. En réduisant le nombre de requêtes par page, on diminue le nombre de ressources serveurs nécessaires.*

#### **Moyens de mise en œuvre :**

- Définir des indicateurs de poids maximum par écran, toutes ressources confondues.
- Définir des indicateurs de nombre de requêtes client/serveur maximum par écran. Par exemple, viser moins de 30 requêtes par page.
- Utiliser des techniques comme les spritesheets CSS pour regrouper plusieurs petites images en une seule, ce qui réduit le nombre de requêtes HTTP.

#### **Critères de validation :**

- Le poids maximum par écran est défini et respecté. Ce poids doit être proportionné et affiché dans la déclaration d'écoconception.
- Le nombre de requêtes maximum par écran est défini et respecté. Il faut préciser si cette limitation concerne uniquement le chargement initial de la page ou aussi son fonctionnement (par exemple, les requêtes de contrôle de champs dans un formulaire).

#### **Liens :**

- [https://www.arcep.fr/uploads/tx\\_gspublication/referentiel\\_general\\_ecoconception\\_des\\_services\\_numeriques\\_version\\_2024.pdf#page=93](https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=93)
- Les 115 bonnes pratiques d'écoconception web : 46



*Sprint Planning / Frontend / Optimisation des Ressources*

*Complexité : 2 – Priorité : 2*

*Il n'est pas utile de charger des images qui ne sont pas encore visibles à l'écran, on peut alors utiliser alors le "lazy loading" ou chargement paresseux pour ne charger que les images, vidéos, et autres médias lorsqu'ils deviennent visibles à l'écran. Résultat : moins de données consommées et une navigation plus fluide.*

**Moyens de mise en œuvre :**

- Ne chargez les ressources et composants que lorsqu'ils sont effectivement utilisés. Utilisez des mécanismes de chargement progressif pour les éléments graphiques et les médias. Par exemple, pour les vidéos, pensez au streaming. Pour les images, utilisez le "lazy loading" pour ne charger que celles qui sont affichées à l'écran lorsque l'utilisateur fait défiler la page.

```

```

- Idem pour les iframes, utilisez l'attribut `loading="lazy"` pour ne charger les iframes que lorsqu'elles deviennent visibles à l'écran.

Exemple :

```
```html
```

```
<iframe src="video-player.html" title="..." loading="lazy"></iframe>
```

```
```
```

**Critères de validation :**

- Toutes les images, iframes et vidéos sont chargés avec lazy loading.

**Liens :**

- [https://www.arcep.fr/uploads/tx\\_gspublication/referentiel\\_general\\_ecoconception\\_des\\_services\\_numeriques\\_version\\_2024.pdf#page=98](https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=98)
- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=34>
- Les 115 bonnes pratiques d'écoconception web : 50

*Sprint Planning / Frontend / Optimisation des Ressources*

*Complexité : 3 – Priorité : 3*

*Le repaint et le reflow sont deux opérations qui peuvent sérieusement solliciter les ressources CPU lors du rendu d'une page web. Le repaint se produit lorsque l'apparence d'un élément du Document Object Model (DOM) change, sans que cela n'affecte la structure de la page. En revanche, le reflow, qui est souvent plus coûteux en termes de ressources, se produit lorsque la position ou la taille des éléments dans le DOM doit être recalculée.*

**Moyens de mise en œuvre :**

Pour réduire les repaint, il est conseillé de ne pas toucher aux propriétés stylistiques d'un élément, telles que la couleur de fond, le style de bordure, la couleur du texte ou la taille. En ce qui concerne le reflow, il faut limiter les modifications apportées aux propriétés qui influencent la position ou la dimension des éléments, ainsi que le type de positionnement ou le contenu. Cela est particulièrement vrai pour certains éléments HTML, comme les tables, dont le reflow peut nécessiter un temps de traitement jusqu'à 3x plus long que pour des éléments avec un affichage de type block. Pour aller plus loin : <https://developers.google.com/speed/articles/reflow>.

**Critères de validation :**

- Les modifications apportées aux éléments du DOM sont principalement limitées à celles qui n'entraînent pas de reflow.
- Les outils de développement du navigateur indiquent un faible nombre de repaint et de reflow lors de l'analyse des performances de la page.

**Liens :**

- Les 115 bonnes pratiques d'écoconception web : 51

*Sprint Planning / Frontend / Optimisation des Ressources*

*Complexité : 2 – Priorité : 3*

*L'élément HTML canvas est principalement utilisé pour dessiner des graphiques, créer des jeux ou générer des images dynamiques grâce à des API JavaScript. Cependant, son utilisation peut entraîner une sollicitation importante des ressources CPU et GPU, ce qui peut ralentir les appareils moins performants. Il est donc conseillé d'explorer d'autres éléments HTML natifs lorsque cela est possible.*

**Moyens de mise en œuvre :**

- Effectuer les rendus uniquement lorsque cela est nécessaire.
- Pour les animations, privilégier l'utilisation de `window.requestAnimationFrame()` au lieu de `window.setInterval()`, car cela permet une gestion plus efficace des animations en synchronisant le rendu avec le taux de rafraîchissement de l'écran.
- Éviter de recourir à des bibliothèques lourdes qui pourraient ajouter une charge inutile au processus de rendu.
- Tester régulièrement les performances de rendu pour identifier les goulots d'étranglement et optimiser le code en conséquence.

**Critères de validation :**

- Les rendus sont effectués uniquement lorsque cela est nécessaire, sans appels superflus.
- Les animations utilisent `window.requestAnimationFrame()`.
- Les bibliothèques utilisées sont légères et adaptées aux besoins du projet.

**Liens :**

- Les 115 bonnes pratiques d'écoconception web : 54

*Sprint Planning / Frontend / Optimisation JavaScript*

*Complexité : 3 – Priorité : 3*

*L'utilisation d'un linter est essentielle pour s'assurer que le code que vous écrivez (que ce soit en CSS, JavaScript, HTML, etc.) est correct sur le plan syntaxique et respectueux de vos conventions de codage. En intégrant des outils de validation de code dans votre processus de livraison, vous améliorez la qualité de votre projet et réduisez les risques de bugs ou de comportements inattendus. Certaines règles de ce référentiel peuvent être ajoutés pour être surveillées automatiquement.*

**Moyens de mise en œuvre :**

Utilisez des outils comme :

- ESLint pour le code JavaScript, qui vous aide à identifier et à corriger les problèmes de syntaxe et de style dans votre code. <https://eslint.org/>
- Stylelint pour vos feuilles de styles, qui permet de s'assurer que votre CSS respecte les bonnes pratiques et les conventions de style. <https://stylelint.io/>

Ces outils peuvent être intégrés dans votre environnement de développement ou votre pipeline de livraison continue, ce qui vous permet de détecter les erreurs dès leur apparition.

**Critères de validation :**

- Tous les fichiers de code ont été analysés par le linter sans générer d'erreurs.

**Liens :**

- Les 115 bonnes pratiques d'écoconception web : 42

*Sprint Planning / Frontend / Optimisation JavaScript*

*Complexité : 3 – Priorité : 3*

*L'exécution de JavaScript peut monopoliser le processeur du terminal alors que celui-ci était en train de calculer les repaint et reflow, ce qui entraîne des blocages dans l'affichage de l'interface et limite les interactions de l'utilisateur avec la page. En conséquence, l'utilisateur peut ressentir une lenteur dans l'expérience de navigation, ce qui peut le pousser à tous les sites qu'il visite présentent le même comportement à renouveler son appareil en raison d'une obsolescence perçue.*

*Pour remédier à cette situation, il est conseillé de fragmenter le code JavaScript en petites tâches qui s'exécutent à des moments appropriés, plutôt que de tout traiter d'un coup.*

**Moyens de mise en œuvre :**

Suivre le Total Blocking Time (TBT) pour évaluer l'impact des tâches JavaScript sur la performance. Cette dernière mesure le temps total durant lequel des tâches longues (celles qui durent plus de 50 ms) occupent le thread principal du navigateur. Veillez à ce que le TBT reste en dessous de 200 ms pour garantir une expérience utilisateur fluide. Pour effectuer cette mesure, des outils comme LightHouse ou Web Page Test (<https://webpagetest.org/>) peuvent être utilisés.

**Critères de validation :**

- Des outils de mesure tels que LightHouse ou Web Page Test permettent de suivre le Total Blocking Time régulièrement.
- Le Total Blocking Time est mesuré et reste en dessous de 200 ms.

**Liens :**

- Les 115 bonnes pratiques d'écoconception web : 52

*Sprint Planning / Frontend / Optimisation JavaScript*

*Complexité : 3 – Priorité : 3*

*L'utilisation des bases de données clé-valeur comme Indexeddb et Web Storage, ainsi que la mise en cache via l'API Cache Storage, permet de conserver des données persistantes directement sur le navigateur de l'utilisateur. Cette approche réduit le besoin d'effectuer des requêtes répétées au serveur, et accélère la récupération des données stockées localement. En optimisant le stockage des données, on peut également alléger la charge sur les serveurs, ce qui peut conduire ultimement à réduire le nombre d'équipements nécessaires pour gérer le trafic.*

**Moyens de mise en œuvre :**

Utilisez le localStorage via sa variable nativement accessible dans le code javascript :

```
```javascript
localStorage.setItem('name', 'nom_utilisateur');
var myName = localStorage.getItem('name');
```
```

Les Framework javascript proposent souvent un module d'abstraction pour gérer le stockage local. Par exemple, en Vuejs nous utilisons les librairies Vuex et Pinia.

**Critères de validation :**

- Les données statiques avec un faible taux de rafraichissement sont stockées localement.

**Liens :**

- Les 115 bonnes pratiques d'écoconception web : 63
- Les 115 bonnes pratiques d'écoconception web : 71

---

## SP-FE-25 – Utiliser des en-têtes Expires ou Cache-Control pour optimiser le cache

---

*Sprint Planning / Frontend / Optimisation JavaScript*

*Complexité : 2 – Priorité : 2*

L'idée ici est de tirer parti des en-têtes HTTP Expires et Cache-Control pour indiquer aux navigateurs combien de temps ils doivent conserver une ressource en cache. En configurant correctement ces en-têtes pour les fichiers CSS, JavaScript et les images, on peut réduire le nombre de requêtes HTTP envoyées au serveur.

### Moyens de mise en œuvre :

Configurez les en-têtes Expires et Cache-Control sur votre serveur web. Voici un exemple de configuration pour un serveur Apache :

```
``apache
BEGIN Cache-Control Headers
<IfModule mod_headers.c>
<FilesMatch ""\.(\.ico|jpe?g|png|gif|swf|css|gz)$"">
Header set Cache-Control ""max-age=2592000, public""
</FilesMatch>
<FilesMatch ""\.(\.html|htm)$"">
Header set Cache-Control ""max-age=7200, public""
</FilesMatch>
</IfModule>
END Cache-Control Headers
``
```

La stratégie de cache doit être adaptée au contexte de l'application et au type de contenu transféré fréquemment. Par exemple, les images et les fichiers CSS peuvent avoir une durée de vie en cache plus longue que les pages HTML.

### Critères de validation :

- Les en-têtes Expires ou Cache-Control sont en place sur le serveur.
- Un mécanisme de cache côté utilisateur est mis en place lorsque cela est pertinent. Pour tester cela, accédez plusieurs fois au service et vérifiez si les contenus sont récupérés à partir du cache plutôt que du serveur.
- La stratégie de cache frontend est explicitée dans la déclaration d'écoconception du service, en tenant compte du type de contenu, du contexte d'application et des scénarios d'usage.

### Liens :

- Les 115 bonnes pratiques d'écoconception web : 74
- [https://www.arcep.fr/uploads/tx\\_gspublication/referentiel\\_general\\_ecoconception\\_des\\_services\\_numeriques\\_version\\_2024.pdf#page=94](https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=94)

---

## SP-FE-26 – Réduire les accès au DOM via JavaScript

---

*Sprint Planning / Frontend / Optimisation JavaScript*

*Complexité : 3 – Priorité : 3*

*L'accès au DOM via JavaScript est une procédure lourde qui consomme beaucoup de cycles processeur (CPU). Réduisez autant que possible ces accès.*

*L'utilisation de bibliothèques comme le Shadow DOM peut contribuer à cette optimisation, car elles permettent de regrouper les modifications apportées à l'arbre DOM, réduisant ainsi le nombre d'opérations nécessaires. A noter que certains Framework comme Vuejs ou Svelte utilisent le shadow DOM nativement.*

### **Moyens de mise en œuvre :**

Consulter le projet de Shadow DOM proposé par le W3C, accessible à l'adresse suivante : (<https://github.com/WICG/webcomponents>). Ce projet fournit des recommandations précieuses pour réduire les accès au DOM via JavaScript.

Vous pouvez explorer les bonnes pratiques en JavaScript sur la page suivante : [https://www.w3.org/wiki/JavaScript\\_best\\_practices#Keep\\_DOM\\_access\\_to\\_a\\_minimum](https://www.w3.org/wiki/JavaScript_best_practices#Keep_DOM_access_to_a_minimum).

### **Critères de validation :**

- L'utilisation de bibliothèques comme le Shadow DOM est intégrée dans le projet.

### **Liens :**

- Les 115 bonnes pratiques d'écoconception web : 56



---

## SP-FE-27 – Utiliser la délégation d'événements

---

*Sprint Planning / Frontend / Optimisation JavaScript*

*Complexité : 3 – Priorité : 3*

*La délégation d'événements est une technique qui permet de gérer les événements de manière plus efficace en évitant de créer un écouteur d'événements pour chaque élément du Document Object Model (DOM) que l'on souhaite surveiller. Au lieu de cela, on attache un seul écouteur à un élément parent, qui va ensuite gérer les événements pour tous ses enfants. Cela réduit la sollicitation de ressources, car moins de mémoire est utilisée pour stocker ces écouteurs, et cela peut également améliorer les performances de l'application, surtout lorsque le nombre d'éléments à surveiller est élevé.*

### **Moyens de mise en œuvre :**

Identifier un élément du DOM qui servira de délégué. Cet élément doit être capable d'intercepter les événements qui se produisent sur ses enfants. Utiliser la méthode `addEventListener` pour attacher un écouteur d'événements à cet élément délégué. Il vous reviendra ensuite de gérer correctement le traitement à effectuer en fonction de l'élément enfant qui a été sollicité.

### **Critères de validation :**

- L'écoute d'un même événement sur plusieurs éléments HTML, est délégué à un élément parent.

### **Liens :**

- Les 115 bonnes pratiques d'écoconception web : 55

---

## SP-UX-01 – Remplacer les icônes de réseaux sociaux par du fait maison

---

### Sprint Planning / UX/UI / Réseaux sociaux

Complexité : 1 – Priorité : 1

*De nombreux sites web intègrent des liens vers des réseaux sociaux via des plugins qui, bien souvent, nécessitent l'utilisation de libraires Javascript. Cela peut sembler pratique pour gagner du temps sur les développements, mais cela entraîne une sollicitation de réseau supplémentaire qui pourrait être évitée. De plus, ces plugins intègrent des cookies qui permettent le profilage et le suivi des utilisateurs partout où ceux-ci sont installés. Il est pertinent de se demander si la présence de ces icônes sur chaque page est réellement indispensable ou si cela relève davantage d'une habitude que d'un besoin avéré.*

#### Moyens de mise en œuvre :

Pour remplacer ces plugins, une solution simple consiste à les remplacer par une version SVG de l'icône accompagnée d'un lien :

```
```html
<a class="nav-link" href="https://www.linkedin.com/company/davidson-consulting">

</a>
```
```

#### Critères de validation :

- Les icônes de réseaux sociaux sont intégrées sous forme d'images avec des liens directs, sans utilisation de plugins Javascript.

#### Liens :

- <https://beta.designersethiques.org/designersethiques.org/media/pages/thematique-ecoconception/guide-d-ecoconception/fb2ce45b86-1703181312/guide-v2-hyperliens.pdf#page=45>

# Sprint Review

*Cérémonie au cours de laquelle l'équipe Scrum présente l'incrément de produit réalisé durant le sprint aux parties prenantes, recueille leurs retours et ajuste le backlog produit en conséquence*

---

## SR-FE-01 – Valider les pages auprès du W3C

---

### *Sprint Review / Frontend / W3C*

*Complexité : 2 – Priorité : 3*

*Un code HTML bien formé permet une meilleure accessibilité, un meilleur référencement et une amélioration des performances. Dans le cas contraire, le navigateur corrigera dynamiquement un certain nombre d'éléments pour afficher au mieux les pages posant des problèmes. Ces corrections dynamiques consomment inutilement des ressources à chaque chargement des pages concernées.*

#### **Moyens de mise en œuvre :**

Utiliser le validateur du W3C (World Wide Web Consortium) pour vérifier que les pages sont bien valides et que le code HTML est correctement formé : <https://validator.w3.org> ou <https://validator.w3.org/nu/> pour filtrer les erreurs.

#### **Critères de validation :**

- Le code HTML est conforme aux spécifications du W3C.
- Les pages se chargent sans erreurs de rendu dans les principaux navigateurs.

#### **Liens :**

- Les 115 bonnes pratiques d'écoconception web : 60

---

## SR-SP-01 – Mener des revues de conception et des revues de code

---

### *Sprint Review / Spécifications / Revues d'écoconception*

*Complexité : 2 – Priorité : 2*

*Pour parvenir à une solution optimale en termes de sobriété numérique, il faut intégrer dès le départ l'objectif de minimiser la sollicitation de ressources.*

*Cependant, vérifier régulièrement la validité des choix de conception au regard de leurs impacts est essentiel pour s'assurer que le système initialement bien conçu ne s'alourdisse par une accumulation de choix néfastes.*

#### **Moyens de mise en œuvre :**

Assurer le respect de l'écoconception du service via :

- La revue de code afin de détecter les patterns de développement qui alourdissent le système.
- La revue d'écoconception prenant en compte l'impact des fonctionnalités dans la sollicitation de ressources. L'équipe projet doit être en mesure de définir les impacts directs et indirects pour chaque fonctionnalité.
- L'impact poker est un excellent outil pour cela : <https://www.suricats-consulting.com/impact-poker-agile-engage/>

#### **Critères de validation :**

- Des revues de code sont régulièrement effectuées.
- Des revues d'écoconception sont régulièrement effectuées.

#### **Liens :**

- [https://www.arcep.fr/uploads/tx\\_gspublication/referentiel\\_general\\_ecoconception\\_des\\_services\\_numeriques\\_version\\_2024.pdf#page=44](https://www.arcep.fr/uploads/tx_gspublication/referentiel_general_ecoconception_des_services_numeriques_version_2024.pdf#page=44)

# ANNEXES

## La déclaration d'écoconception

Version Excel avec calcul de score d'avancement : [Lien vers le site de l'ARCEP](#)

Version word : [Lien vers le site de l'ARCEP](#)

Exemple d'une déclaration d'écoconception version Web : [Lien vers le site de la DINUM](#)

La publication d'une déclaration d'écoconception est un prérequis pour se prévaloir de l'application du référentiel général de l'écoconception des services numérique (RGESN). L'objectif est de garantir la transparence concernant les efforts d'écoconception du service, tout en pouvant les valoriser dans un document accessible au public.

Cette déclaration est mentionnée à de nombreuses reprises dans ce référentiel. Elle peut être circonscrite au service numérique évalué ou s'inscrire dans une publication plus générale, par exemple à l'échelle de l'organisation dans une démarche plus globale de numérique soutenable. La déclaration d'écoconception doit être datée et régulièrement mise à jour (idéalement à chaque modification significative du service).

Le document peut être publié sur le site internet du service numérique. Selon sa forme, cette publication peut être affichée par exemple dans les mentions légales, aux côtés de la déclaration d'accessibilité ou sur une page dédiée. Cette déclaration volontaire doit revenir sur le détail de la mise en œuvre des critères du référentiel général de l'écoconception des services numériques applicables au service.

## Les ressources intéressantes

- Impact Poker Agile - Suricats: <https://www.suricats-consulting.com/impact-poker-agile-engage/>
- Kastor.green: <https://kastor.green/>
- RGAA: <https://accessibilite.numerique.gouv.fr/>
- Plugin Lighthouse: <https://developer.chrome.com/docs/lighthouse/overview?hl=fr>
- PageSpeed Insights: <https://pagespeed.web.dev/>
- WebPageTest: <https://www.webpagetest.org/>
- GreenIT-Analysis-cli: <https://github.com/cnumr/GreenIT-Analysis-cli>
- Jamstack.org: <https://jamstack.org/>
- Guide pratique pour des achats numériques responsables: <https://ecoresponsable.numerique.gouv.fr/publications/guide-pratique-achats-numeriques-responsables/>
- WebspeedTest: <https://webspeedtest.cloudinary.com/>
- Déclaration d'écoconception – Dinum : <https://ecoresponsable.numerique.gouv.fr/declaration-ecoconception/>
- Green Software Foundation: <https://greensoftware.foundation/>
- Méthodologie d'évaluation environnementale basée sur l'Analyse du Cycle de Vie (ACV) pour un service numérique: <https://librairie.ademe.fr/produire-autrement/6022-referentiel-par-categorie-de-produit-rcp-des-services-numeriques.html>
- World Electricity Map: <https://app.electricitymaps.com/map/72h/hourly>