

keyrus
make data matter



Le pari du code décarboné

Vers une programmation éco-responsable

Le pari du code décarboné

Vers une programmation éco-responsable

Véritable reflet d'une prise de conscience croissante de l'impact écologique des technologies numériques, le concept de code décarboné s'inscrit désormais au cœur des préoccupations du secteur digital et numérique.

Le code décarboné, par définition, vise à réduire l'empreinte carbone associée au développement et à l'exécution des logiciels et plateformes.

Et en théorie cette approche s'intéresse non seulement à l'efficacité immédiate de l'exécution des applications, en termes de consommation réduite de CPU et de RAM, mais aussi à l'ensemble du processus de développement du code, y compris l'utilisation des technologies d'intelligence artificielle (IA).

En effet, pour évaluer l'impact réel du code décarboné, une approche holistique est nécessaire. Il ne s'agit pas seulement de mesurer la performance environnementale du code lui-même, mais aussi de prendre en compte les ressources utilisées pour la création, le calcul, le stockage, et même les équipes de développement.

Ainsi, aussi fédérateur soit-il, ce concept soulève tout de même des interrogations complexes, qui questionnent la contradiction entre quête des performances et la consommation raisonnée des ressources. Quelles et où sont les véritables avancées écologiques ?

Afin de démêler les multiples facettes du code décarboné et d'évaluer son véritable impact, il paraît essentiel d'approfondir l'exploration de ces pistes. Mieux éclairer pour mieux comprendre, maîtriser et agir.

Images générées par Midjourney avec la consigne : « Portrait of a computer programmer in front of a digital landscape showing a futuristic city and lines of code, with a lush green environment merging, symbolizing the harmonious integration of sustainable coding practices and advanced technology. Picture could include computers and servers, databases. »



Enjeux du code décarboné

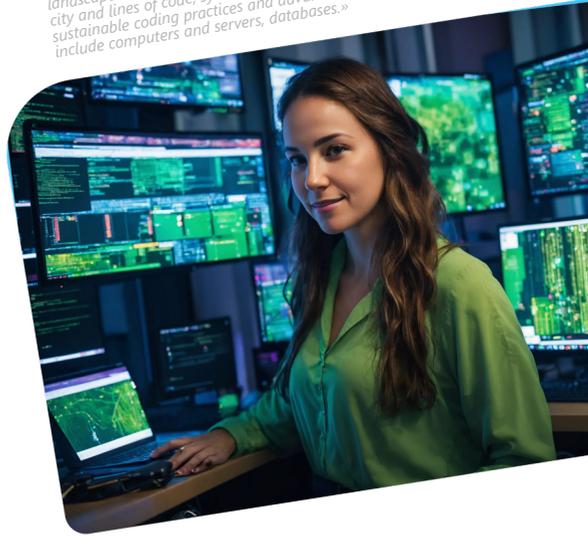
Les enjeux associés au code décarboné dans le secteur informatique ou digital sont à la fois vastes et significatifs, touchant divers aspects allant de l'environnement à la responsabilité sociale des entreprises.

L'un des enjeux majeurs du code décarboné est la **réduction de l'impact environnemental** lié au développement et à l'exécution des logiciels. Cette démarche vise à minimiser l'empreinte carbone, répondant ainsi aux préoccupations croissantes en matière de changement climatique et de durabilité environnementale. Pour les entreprises, l'adoption de pratiques de développement décarbonées peut non seulement contribuer à la protection de l'environnement, mais aussi améliorer leur image et leur responsabilité sociale d'entreprise, un facteur de plus en plus important pour les consommateurs et les investisseurs.

Parallèlement, le code décarboné stimule **l'innovation technologique**. Il encourage le développement de nouvelles méthodes et de technologies plus efficaces et moins consommatrices en énergie. Cette innovation technologique offre aux entreprises l'opportunité de se positionner à la pointe du progrès, tout en bénéficiant de gains d'efficacité et potentiellement de réductions de coûts à long terme.

Un autre enjeu important est la nécessité de maintenir un équilibre entre la réduction de l'empreinte carbone et la **performance des logiciels**. Les entreprises doivent veiller à ce que les initiatives de code décarboné n'entravent pas la qualité ou l'efficacité des applications. La performance reste un critère essentiel pour la satisfaction client et la compétitivité sur le marché. Trouver cet équilibre est donc fondamental pour assurer le succès commercial tout en adhérant aux principes de durabilité.

Image générée par Lexica Aperture v3.5, avec la consigne (prompt) : « Portrait of a woman computer programmer in front of a digital landscape showing a lush green environment merging with a futuristic city and lines of code, symbolizing the harmonious integration of sustainable coding practices and advanced technology. Picture could include computers and servers, databases.»



Enfin, **l'adoption à l'échelle** de l'industrie du code décarboné est un enjeu majeur. Pour que son impact soit significatif, il est essentiel que ces pratiques soient largement adoptées et intégrées dans les normes industrielles. Cela implique un engagement non seulement des entreprises individuelles, mais aussi des acteurs de l'industrie dans son ensemble. Une adoption généralisée peut mener à des changements substantiels dans la manière dont l'industrie IT opère, positionnant le secteur comme un leader en matière de pratiques durables.

Ces enjeux soulignent l'importance stratégique du code décarboné pour les entreprises du secteur informatique. Non seulement ces pratiques peuvent aider à atténuer l'impact environnemental de l'industrie, mais elles offrent également des opportunités de développement et d'innovation, tout en alignant les entreprises avec les valeurs de durabilité et de responsabilité sociale de plus en plus valorisées sur le marché actuel.

“ Non seulement les bonnes pratiques peuvent aider à **atténuer l’impact environnemental** de l’industrie, mais elles offrent également **des opportunités de développement et d’innovation.** ”

Image générée par Midjourney, avec la consigne :
« Green computer coding »



Les défis de la production décarbonée

Toute la démarche du code décarboné s'inscrit donc dans une tendance plus large de la technologie verte, qui cherche à concilier progrès technologique et durabilité. À travers le code décarboné, l'industrie informatique et/ou digitale reconnaît son rôle et sa responsabilité dans la lutte contre le changement climatique. Cela va au-delà de la simple optimisation du code ; **c'est une réflexion profonde sur la manière dont la technologie peut être conçue et utilisée de manière écologiquement responsable.**

Cependant, la mise en œuvre de cette démarche soulève plusieurs défis. Comme nous l'avons déjà évoqué, l'un des plus importants est **l'équilibre entre la performance et la durabilité.** Les entreprises se trouvent souvent face à la difficulté de maintenir des performances élevées tout en réduisant l'empreinte carbone de leurs produits logiciels.

Par exemple, l'efficacité est souvent recherchée à travers l'accélération de la production de code, facilitée par l'emploi d'outils d'intelligence artificielle avancés. Cependant, ces outils d'IA peuvent être eux-mêmes gourmands en ressources, créant ainsi un paradoxe où l'outil censé optimiser le code contribue indirectement à une augmentation de l'empreinte carbone.

Certains observateurs du secteur IT s'interrogent même sur le fait de savoir si cette démarche représente un véritable progrès écologique ou si elle relève davantage d'une stratégie de greenwashing. Par exemple, est-ce que la réduction de l'utilisation de la CPU et de la RAM pour un logiciel donné compense suffisamment l'énergie et les ressources dépensées dans le développement de solutions plus complexes, mais potentiellement plus écologiques ? En d'autres termes, le code décarboné représente-t-il un équilibre optimal entre l'efficacité technique et la responsabilité environnementale, ou est-il simplement une extension des pratiques de développement existantes, rebaptisées sous une nouvelle terminologie plus écologique ?

Ce défi nécessite non seulement des avancées technologiques, mais aussi **un changement de mentalité dans la conception et le développement des logiciels.**

Un autre défi majeur est le développement et **la mise en place de standards et de méthodologies** pour mesurer l'impact environnemental du code. Actuellement, l'industrie manque de normes uniformes pour évaluer et comparer l'empreinte carbone des pratiques de développement, ce qui rend difficile pour les entreprises d'évaluer l'efficacité de leurs efforts de décarbonation.



Images générées par **Lexica Aperture v3.5**, avec la consigne (prompt) : « *Portrait of a woman computer programmer in front of a digital landscape showing a lush green environment merging with a futuristic city and lines of code, symbolizing the harmonious integration of sustainable coding practices and advanced technology. Picture could include computers and servers, databases.* »

Cela implique de former les développeurs et les équipes de gestion de projet aux nouvelles méthodes et outils, et de changer les habitudes de développement établies.

L'intégration du code décarboné dans les processus de développement existants est aussi un défi. Les entreprises doivent revoir leurs cycles de vie de développement logiciel et adapter leurs équipes et leurs processus pour intégrer les considérations de durabilité, ce qui peut être un défi organisationnel et technique.

Enfin, **mesurer et quantifier les bénéfices** du code décarboné reste un défi. Déterminer l'efficacité réelle des mesures de décarbonation en termes de réduction de l'empreinte carbone est complexe et nécessite des outils de mesure précis et une analyse continue. Mais là encore, l'émergence depuis quelques mois de multiples solutions spécialisées dans ce genre de mesure et d'analyse devrait rapidement changer la donne.

On l'aura compris, toutes les complexités de l'intégration des pratiques de développement durable sont en train de tomber les unes après les autres. Le momentum digital semble aujourd'hui à son apogée, avec toutes ses promesses d'opportunités et d'innovation dans la transition vers un développement plus responsable et un avenir plus durable.

Mais une fois qu'on a listé ces défis, que fait-on concrètement ? Quelles sont les bonnes pratiques à suivre et à encourager ?

La gestion des coûts et des ressources nécessaires pour adopter des pratiques de code décarboné représente également un obstacle. Pour de nombreuses entreprises, en particulier les PME et les startups, les coûts initiaux et les investissements en ressources humaines et technologiques pour le développement de code décarboné peuvent être prohibitifs. Notons néanmoins que ce phénomène aurait plutôt tendance à s'inverser depuis quelques mois, avec l'apparition de plus en plus de solutions d'IA as a service abordables et plus facilement accessibles aux organisations plus agiles en raison de leur taille modeste ou intermédiaire.

En outre, **la sensibilisation et la formation** sont essentielles pour surmonter les résistances au changement et pour intégrer les principes de durabilité dans la culture d'entreprise.

Images générées par Midjourney, avec la consigne (prompt) :
« Rational use of server resources.»

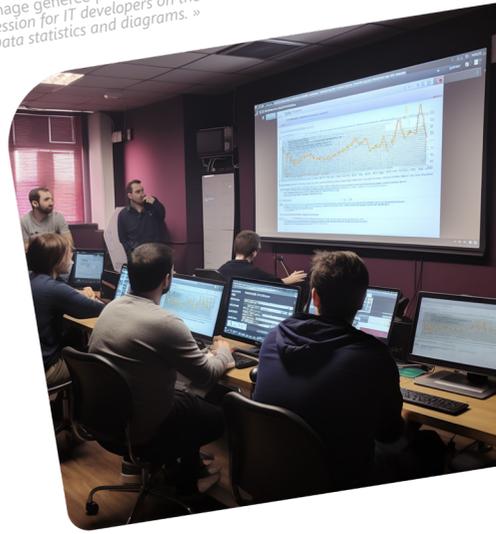
Les bonnes pratiques de développement

Les bonnes pratiques de développement de code décarboné visent à réduire l'empreinte carbone et la consommation de ressources des logiciels. Voici quelques-unes de ces pratiques clés :

- 1. Optimisation du code :** écrire un code propre et efficace qui exécute des tâches avec le moins de ressources possibles. Cela inclut l'élimination du code redondant ou inutile, la réduction de la complexité algorithmique et l'optimisation des boucles et des requêtes, ainsi que la gestion efficace de la mémoire.
- 2. Gestion éco-efficace des données :** utiliser les bases de données et les structures de données de manière efficace pour réduire les besoins en stockage et en traitement. Cela peut inclure des techniques comme la compression de données, l'utilisation de formats de données efficaces et l'indexation appropriée.
- 3. Utilisation rationnelle des ressources serveur :** optimiser l'utilisation des serveurs, notamment en termes de gestion de la charge et de l'équilibrage, pour réduire la consommation d'énergie (un peu comme l'EDF qui jadis régulaît la consommation avec des heures creuses et des heures pleines). L'adoption de l'infrastructure en tant que service (IaaS) peut également contribuer à une utilisation plus efficace des ressources.
- 4. Développement basé sur le cloud :** tout en proposant une gestion plus efficace des ressources, doit être considéré avec nuance lorsqu'il s'agit de son impact environnemental comparé aux solutions on-premise. Bien que les fournisseurs de cloud offrent souvent des infrastructures conçues pour minimiser la consommation d'énergie, l'avantage écologique de ces services n'est pas universel et dépend de plusieurs facteurs. L'infrastructure existante de l'entreprise et les besoins spécifiques du projet jouent un rôle crucial dans cette évaluation. Dans certains cas, des solutions alternatives, telles que des mini bases de données locales sur un appareil mobile ou l'utilisation de caches, peuvent se révéler moins gourmandes en énergie que le déploiement de solutions entièrement basées sur le cloud.
- 5. Éco-conception :** intégrer les considérations environnementales dès les premières phases de conception des logiciels. Cela inclut la planification de l'architecture logicielle en tenant compte de son impact écologique.

- 6. Tests et profilage énergétique** : effectuer des tests réguliers pour évaluer l'efficacité énergétique du logiciel. Utiliser des outils de profilage pour identifier et corriger les parties du code qui consomment le plus d'énergie.
- 7. Utilisation de langages de programmation efficaces** : choisir des langages de programmation et des frameworks qui sont connus pour leur efficacité énergétique, ou qui offrent des outils et des bibliothèques pour aider à réduire la consommation de ressources. Certains langages de programmation, tels que **Rust**, sont spécifiquement conçus pour offrir une performance accrue tout en étant plus économes sur le plan énergétique. Concernant l'utilisation de bibliothèques optimisées, plusieurs d'entre elles ont été développées avec un focus sur l'efficacité environnementale. **TensorFlow Lite for Microcontrollers**, par exemple, est une version allégée de **TensorFlow**, la célèbre bibliothèque d'apprentissage automatique, optimisée pour les microcontrôleurs. **React Native** se distingue par des performances élevées et une utilisation efficiente de l'énergie des appareils, grâce à son architecture et à son rendu natif. D'autres outils tels que **GreenSock Animation Platform (GSAP)**, **Flutter**, **D3.js**, **Flask**, et **Elm** sont également notables pour leur efficacité.

Mais attention, là aussi il est important de noter que malgré les faibles différences entre les langages de programmation en termes d'utilisation nominale, les études se concentrent souvent sur des fonctions avancées qui sont rarement, voire jamais, utilisées dans le développement courant. De plus, ces études ne prennent pas toujours en compte les exigences d'installation. Par exemple, la comparaison entre .NET, qui nécessite toute une suite d'installations, et Java avec sa JVM [Java Virtual Machine] associée, par rapport à Python, qui est presque autonome. Pourtant, dans de telles études, Java et .NET sont souvent considérés comme étant plus performants que Python.



- 8. Formation et sensibilisation des développeurs** : former les développeurs aux principes et techniques de code décarboné. Cela inclut la sensibilisation aux impacts environnementaux de leurs pratiques de codage. On notera ici qu'il peut y avoir un phénomène générationnel, les développeurs plus jeunes étant plus ouverts ou « formatés » au principe de code décarboné que les développeurs, techniciens et ingénieurs plus expérimentés.
- 9. Suivi continu et amélioration** : mettre en place des mécanismes de suivi pour évaluer continuellement la performance environnementale des logiciels et apporter des améliorations au fil du temps.
- 10. Documenter et partager les bonnes pratiques** : documenter les stratégies et les techniques utilisées pour le développement de code décarboné et partager ces connaissances au sein de la communauté de développeurs pour encourager une adoption plus large.

En adoptant ces pratiques, les développeurs et les entreprises peuvent non seulement contribuer à réduire l'impact environnemental de leurs logiciels, mais aussi souvent réaliser des économies en termes de coûts de fonctionnement et d'infrastructure. Mais si on en revient à l'impact, la question fondamentale reste celle de la mesure, c'est-à-dire, de son exhaustivité ?

“ L'adoption d'**une vision holistique** est indispensable pour évaluer l'impact réel du code et de l'empreinte carbone du processus de développement logiciel. ”

Pourquoi mesurer l'impact de manière holistique ?

L'adoption d'une vision holistique pour évaluer l'impact réel du code décarboné est essentielle pour plusieurs raisons :

Le développement de logiciels ne se limite pas à l'écriture de code ; il englobe également l'infrastructure sur laquelle ce code est exécuté, les outils utilisés pour le développement, et même les pratiques de travail des équipes de développement.

Une vision holistique permet d'évaluer l'impact environnemental de tous ces aspects, offrant ainsi une image complète de l'empreinte carbone du processus de développement logiciel. Cela devient d'autant plus pertinent avec la mise en œuvre imminente de la Corporate Sustainability Reporting Directive (CSRD) de l'Union Européenne, qui va imposer des réglementations plus strictes sur les entreprises cotées dès 2024. La CSRD vise à augmenter la transparence en matière de durabilité, obligeant les entreprises à fournir des informations détaillées sur la manière dont elles gèrent les enjeux sociaux et environnementaux. Cela signifie que les entreprises devront non seulement mesurer, mais aussi divulguer leur impact environnemental de manière exhaustive.

Il est important d'**équilibrer la performance, la fonctionnalité et la durabilité**. Par exemple, un code peut être très efficace en termes de ressources lors de l'exécution, mais si son développement nécessite une quantité excessive de ressources ou d'énergie, l'avantage global pourrait être diminué.

Une approche holistique aide à **identifier les domaines où des améliorations peuvent être apportées** pour augmenter l'efficacité énergétique à tous les niveaux du processus de développement, de l'architecture logicielle au choix des outils et technologies utilisés.

Images générées par Midjourney, avec la consigne :
« Accurately quantifying the carbon footprint of software development. »



En adoptant une perspective globale, toutes les parties prenantes, des développeurs aux chefs de projet et aux décideurs, peuvent être sensibilisées à leur impact environnemental et encouragées à adopter des pratiques plus durables. La CSRD renforce cette nécessité en rendant la divulgation de l'impact environnemental non seulement une bonne pratique, mais aussi une exigence légale pour les entreprises concernées.



Image générée par DALL-E, avec la consigne (prompt) :
« Evaluate the environmental impact of the code at each stage of its life cycle, from design to maintenance and end of life. Photo realistic image aspect ratio 16/12 »

Comment mesurer l'impact ?

La quantification précise de l'empreinte carbone du développement logiciel paraît indispensable pour évaluer l'efficacité des initiatives de code décarboné et pour guider les futures stratégies de développement durable dans le secteur IT.

On ne le répètera jamais assez, l'un des principaux défis réside dans la complexité inhérente à la mesure de l'impact environnemental du développement logiciel. Cela inclut non seulement l'évaluation de l'empreinte carbone directe liée à l'utilisation des ressources informatiques, mais aussi l'impact indirect associé aux processus de développement, tels que l'énergie consommée par les équipes de développement et les infrastructures de support.

Ainsi pour mettre en œuvre une évaluation holistique de l'impact du code décarboné, plusieurs actions peuvent être entreprises :

- **Analyse du cycle de vie** : évaluer l'impact environnemental du code à chaque étape de son cycle de vie, depuis la conception jusqu'à la maintenance et la fin de vie.
- **Utilisation d'outils de mesure et de suivi** : employer des outils pour mesurer l'utilisation des ressources et l'empreinte carbone à différentes étapes du développement logiciel.

- **Intégration des évaluations environnementales** : inclure des évaluations environnementales régulières dans le processus de développement pour identifier les pratiques à haute intensité énergétique.
- **Collaboration avec des experts en durabilité** : travailler avec des consultants ou des organisations telles que Keyrus, spécialisées en durabilité pour obtenir des conseils sur les meilleures pratiques et les stratégies d'amélioration.
- **Rapports et documentation** : créer des rapports détaillés sur l'empreinte carbone et partager les pratiques réussies pour promouvoir une culture de transparence et d'amélioration continue.

Pour mesurer l'impact environnemental du développement logiciel, y compris dans le cadre du code décarboné, il est important d'utiliser des outils adaptés qui peuvent fournir des données précises et utiles. Voici quelques exemples d'outils qui peuvent être utilisés dans ce contexte :

- **Outils de profilage énergétique** : des outils comme **Intel VTune Profiler** ou **AMD uProf** aident à identifier les parties du code qui consomment le plus d'énergie, permettant aux développeurs d'optimiser ces sections pour réduire l'empreinte carbone globale.
- **Logiciels de gestion de l'énergie** : des solutions comme **EnergyCAP SmartAnalytics** offrent des fonctionnalités avancées pour surveiller la consommation d'énergie des applications et des systèmes informatiques. Cette plateforme fournit une analyse complète de l'efficacité énergétique des projets logiciels, permettant aux entreprises de mieux comprendre et optimiser leur consommation énergétique.
- **Outils de suivi des ressources cloud** : pour les applications basées sur le cloud, des outils comme **AWS CloudWatch** ou **Google Cloud's Operations Suite** permettent de suivre l'utilisation des ressources, ce qui est crucial pour évaluer l'efficacité énergétique dans un environnement cloud.

- **Analyseurs de performance web** : des outils tels que **Lighthouse** de Google ou **WebPageTest** fournissent des analyses détaillées sur la performance des pages web, y compris des aspects liés à l'efficacité énergétique, comme le temps de chargement et l'optimisation des ressources.
- **Outils d'analyse de bases de données** : des logiciels comme **SolarWinds Database Performance Analyzer** aident à surveiller et optimiser les performances des bases de données, réduisant ainsi la charge sur les serveurs et l'empreinte carbone associée.
- **Plateformes d'analyse d'impact environnemental** : des plateformes comme **Ecochain** ou **Fruggr** offrent des solutions pour analyser l'impact environnemental global des entreprises, incluant les aspects liés au développement logiciel.

- **Outils de surveillance de l'infrastructure IT** : des solutions comme **Nagios** ou **Zabbix** permettent de surveiller en temps réel l'utilisation des ressources IT, aidant les entreprises à identifier les inefficacités et à les corriger pour réduire leur empreinte carbone.

En adoptant ces approches et de tels outils, les entreprises peuvent non seulement évaluer de manière plus précise l'impact de leurs pratiques de développement logiciel, mais aussi identifier des stratégies pour réduire leur empreinte carbone globale et progresser vers des opérations plus durables.

Quid des *No-code* et *Low-code* ?

Dans la quête du développement le plus respectueux de l'environnement, les méthodes de programmation code, *no-code* et *low-code* se livrent une compétition serrée. La réalité, cependant, est que ces trois approches sont très proches en termes d'empreinte carbone.

Le véritable enjeu n'est pas tant dans la consommation individuelle de ces méthodologies – une analyse qui s'avèrerait fastidieuse et peu fructueuse sur le plan statistique – mais plutôt dans leur infrastructure et utilisation communes.

Ces méthodes partagent des serveurs similaires, sont hébergées dans les mêmes data-centers, et nécessitent des appareils électroniques identiques pour leur utilisation. La différence majeure réside dans la conception de ces outils plutôt que dans leur fonctionnement proprement dit.



Image générée par DALL-E avec la consigne : « Training session for IT developers on the environmental impacts of coding. Data statistics and diagrams. »

Cependant, les outils *no-code* présentent certains avantages indirects qui pourraient les propulser en tête de la course au développement bas carbone. L'un de ces avantages est leur nature mutualisée. Chaque outil, étant la propriété d'une seule entreprise, peut être optimisé pour l'efficacité par cette dernière.

Par exemple, si une plateforme comme **Webflow** adopte une politique d'éco-conception stricte pour ses sites web, l'effet serait amplifié par le volume de sites utilisant cette plateforme, créant ainsi **un impact positif important par effet de cascade**.



Images générées par Midjourney, avec la consigne : « Evolution of mentalities and practices within IT organizations. Photo realistic image of male and female IT developers changing their coding habits. »

De plus, **la nature mutualisée des outils *no-code*** est un levier puissant pour l'implémentation rapide et à grande échelle de pratiques de sobriété énergétique. Cette mutualisation offre une opportunité unique de réduire l'empreinte carbone de manière collective et significative.

En outre, le *no-code* se distingue par sa capacité à accélérer la transition numérique à l'échelle mondiale. En démocratisant l'accès au développement web, il permet à de nombreux acteurs de créer des solutions numériques qui peuvent réduire considérablement leur empreinte carbone tout en limitant leurs coûts et en contournant la pénurie de développeurs.

En conclusion, bien que le *no-code* ne soit pas radicalement différent des autres approches en termes d'empreinte carbone, il s'agit tout de même de code déjà optimisé et qui à ce titre répond au moins en partie à un des challenge du code décarboné qui est l'optimisation du code. Il détient un potentiel significatif pour influencer positivement le climat. Ce potentiel, combiné à son impact sur la transition numérique, en fait un acteur clé dans le mouvement vers un développement plus durable.

Ainsi, malgré les défis, il est possible d'atteindre un équilibre entre performance, efficacité et responsabilité environnementale. En adoptant ces nouvelles approches, les entreprises peuvent réaliser des économies significatives en termes de coûts énergétiques et contribuer de manière proactive à la réduction de l'empreinte carbone du secteur digital / IT. Et surtout, le passage au code décarboné n'est pas seulement une question de changements technologiques, mais implique également une évolution des mentalités et des pratiques au sein des organisations informatiques.

Image générée par Midjourney, avec la consigne : « Evolution of mentalities and practices within IT organizations. Photo realistic image of male and female IT developers changing their coding habits. »

Keyrus spécialiste de l'éco-conception *no-code*

L'art de l'éco-conception est une compétence précieuse dans l'arsenal des spécialistes, particulièrement lorsqu'il s'agit de la conception des services *no-code*. Cette approche vise à influencer positivement le bilan carbone de ces services dès leur phase de conception. L'éco-conception s'appuie sur trois piliers fondamentaux qui allient diverses compétences professionnelles : la **pertinence**, la **frugalité** et la **simplicité**.

La pertinence est liée à la gestion de projet et au *design thinking*. Il s'agit d'évaluer l'utilité réelle du projet, en posant des questions cruciales telles que : Est-ce que ce service est réellement nécessaire ? Sous quelles conditions ? À quel coût ? Cette étape vise à éliminer l'inutile, réduisant ainsi la puissance de calcul nécessaire et, par conséquent, la consommation énergétique et l'empreinte carbone.

La frugalité concerne la mise en œuvre du projet. Elle englobe le choix des outils *no-code*, l'expérience utilisateur (UX) et le contenu hébergé. Par exemple, le poids des images et vidéos sur un site web a un impact direct sur la consommation énergétique. La frugalité vise donc à offrir la meilleure expérience utilisateur tout en maximisant l'efficacité énergétique.

Enfin, **la simplicité combine la pertinence du projet et son exécution.** Elle englobe tout, de la valeur ajoutée de la solution à la conception du service et à son utilisation par l'utilisateur final.

Il est clair que les enjeux climatiques liés au *no-code* sont étroitement liés à ceux du numérique en général. Cette technologie peut être à la fois propre et sobre, à condition d'être gérée par des mains expertes. Le bilan carbone des outils *no-code* dépend donc largement des concepteurs, de leur compréhension des subtilités du numérique, des mécanismes de référencement, des impacts carbone de chaque outil *no-code* et des meilleures pratiques en matière de conception UX et UI.

Chez Keyrus, nos spécialistes sont prêts à accompagner les entreprises dans ce passage au code décarboné, en apportant leur expertise en éco-conception et en développement *no-code* et *low-code*. Nous vous aidons à naviguer dans ces choix technologiques pour optimiser votre bilan carbone et atteindre vos objectifs de développement durable.



Pertinence et potentiel transformateur

En conclusion, l'exploration du concept de code décarboné révèle sa pertinence et son potentiel transformateur dans le secteur informatique et digital. Commenant comme une réponse innovante aux préoccupations environnementales croissantes, le code décarboné s'avère être bien plus qu'une simple tendance. Il incarne une évolution significative dans la manière dont nous concevons, développons et exécutons les logiciels, en mettant l'accent sur la durabilité et la responsabilité écologique.

Il s'agit d'une démarche holistique qui implique de repenser les méthodes de développement logiciel, d'évaluer l'impact environnemental de manière globale et de sensibiliser l'ensemble de la chaîne de valeur IT à ces enjeux. Les défis, notamment en matière de mesure précise de l'impact environnemental et de l'adoption de ces pratiques à grande échelle, sont réels. Cependant, les exemples concrets et les études montrent que des progrès significatifs sont possibles et sont déjà en cours.

À l'avenir, le code décarboné pourrait devenir une norme dans l'industrie du logiciel, guidant non seulement les pratiques de développement, mais aussi influençant les décisions stratégiques au sein des entreprises technologiques.

En somme, le code décarboné n'est pas seulement un objectif en soi, mais aussi un catalyseur de changement, incitant l'ensemble du secteur informatique à réfléchir et à agir de manière plus durable. C'est un voyage vers une technologie plus responsable et un avenir plus vert, où chaque ligne de code compte dans notre quête collective pour un monde plus durable.

Article co-écrit par keyrus, Chat-GPT⁴ et Google Bard



Image générée par Midjourney avec la consigne : « Portrait of a computer programmer in front of a digital landscape showing a lush green environment merging with a futuristic city and lines of code, symbolizing the harmonious integration of sustainable coding practices and advanced technology. Picture could include computers and servers, databases. »

Vous avez trouvé cette lecture utile? Vous souhaitez sûrement lire aussi :



Who killed BI ?

Vers une nouvelle ère de transformation numérique

Dans le monde en perpétuelle évolution de la technologie de l'information, deux domaines se sont particulièrement illustrés ces dernières années : l'Intelligence Artificielle (IA) et la Business Intelligence (BI). Chacun, à sa manière, a révolutionné la manière dont les entreprises traitent, analysent et tirent des enseignements des données. Mais qu'advient-il lorsque ces deux forces convergent ?

L'IA générative, avec ses capacités de création et de modélisation innovantes, promet de transformer profondément le paysage de la BI. Cette transformation soulève une question provocatrice : L'IA générative va-t-elle tuer la Business Intelligence telle que nous la connaissons ?

Plongeons au cœur de cette interaction dynamique et tentons d'explorer les potentialités, les défis et les incertitudes qui en découlent. Attardons-nous sur les avantages potentiels de l'IA générative dans la consolidation des données, la visualisation, le reporting, et d'autres domaines clés de la BI. Analysons également comment cette évolution change radicalement la manière dont les utilisateurs interagissent avec les plateformes et les outils de BI. Enfin, nous essaierons d'évaluer les implications plus larges pour le monde du travail en général et des affaires en particulier, de l'éthique à la transformation des compétences...

Nous sommes peut-être à l'aube d'une ère où les requêtes en langage naturel et les analyses en temps réel supplanteront les modèles prédéfinis de structuration des données, qui sait ? Ou bien cette convergence marquera-t-elle plutôt une collaboration (joyeuse ?) et enrichissante entre l'humain et la machine ? Cet ebook se conçoit comme une exploration divertissante mais néanmoins approfondie pour comprendre et se forger une opinion, loin des penchants.



SCAN ME

keyrus

make data matter

Acteur international du conseil et des technologies, Keyrus a pour mission de donner du sens aux données, en révélant toute leur portée, notamment sous un angle humain.

Parce que ce ne sont pas tant les données elles-mêmes qui importent, mais les opportunités que nous pouvons développer en les apprivoisant vraiment, nous nous efforçons constamment de comprendre les objectifs que nos clients souhaitent atteindre. Nous explorons et mesurons les comportements, nous les comprenons et les traduisons en un résultat concret. Nous donnons un sens aux réalités que les données portent afin d'aider nos clients à prendre des décisions plus efficaces.

Les données, qu'elles soient grandes, petites, humaines, complexes, historiques ou prospectives, n'ont de sens que lorsqu'elles sont utilisées pour développer les expériences, affiner la compréhension du quotidien et prendre les meilleures décisions.

Notre proposition de valeur est fondée sur cinq grands groupes de services, chacun comprenant des offres multiples :

- **Automatisation et intelligence artificielle** : nous fournissons à nos clients les moyens d'améliorer leur productivité et leur précision sur l'ensemble de leurs processus, afin de se concentrer sur le travail à plus forte valeur ajoutée.
- **Expérience numérique centrée sur l'humain** : la relation avec les clients et l'engagement des collaborateurs constituent deux des plus grands contributeurs au succès global des entreprises. Nous aidons les entreprises à imaginer et à créer des expériences numériques multimodales et fluides pour atteindre leurs objectifs.
- **Mise en œuvre des données et des analyses** : les données sont une clé incontestable du succès pour les entreprises. Lorsqu'elles sont utilisées intelligemment, elles ouvrent des opportunités uniques pour faire face aux défis actuels et futurs. Nous permettons aux organisations de déployer tout le potentiel de leurs données : nous mettons la science des données au profit du développement de l'entreprise.
- **Cloud et sécurité** : le Cloud et les plateformes numériques ont le potentiel de révolutionner la façon dont les données sont transformées en valeur, tout en portant l'extensibilité et la flexibilité à un niveau supérieur. Nous sécurisons l'ensemble de vos données et veillons à ce qu'elles soient protégées et confidentielles.
- **Transformation et innovation** : pour prospérer dans l'écosystème actuel, chaque entreprise doit non seulement accélérer sa transformation numérique, mais aussi acquérir des compétences pour stimuler son adaptabilité, sa résilience et sa compétitivité. Nous aidons nos clients à se transformer avec succès pour développer un meilleur futur.

S'appuyant sur l'expérience cumulée de plus de 3 500 collaborateurs et présent dans 27 pays sur 4 continents, Keyrus est l'un des principaux experts internationaux en matière de données, de conseil et de technologie.

Pour en savoir plus : www.keyrus.fr

Jean-Philippe CLAIR

Directeur Marketing, Communication & Innovation Keyrus

jean-philippe.clair@keyrus.com