Engineer

Portfolio

Des prestations d'ingénierie illustrées par des travaux réalisés.

Engineer ApS a été immatriculée en 2025. La plupart des réalisations présentées ici ont été livrées auparavant par notre fondatrice au sein d'équipes clientes ; certaines ont été réalisées par l'entreprise elle-même, et d'autres depuis dans le cadre d'un emploi salarié. Les mêmes mains et les mêmes exigences — nous précisons chaque contexte plutôt que de les confondre.

Découvrez les réalisations **dans leur contexte** — les décisions, la réalisation et les résultats rapportés.

Voir l'ensemble du catalogue

Trouvez une expérience utile à mettre en pratique

Utilisez ce parcours pour trouver une expérience pertinente pour votre projet.

Découvrir nos prestations Contactez-nous →

Lisez les décisions derrière les réalisations

Les entrées décrivent des travaux livrés dans différentes prestations. Lisez une analyse pour découvrir la situation et les décisions qui les sous-tendent.

Études de cas Contactez-nous →

Ces noms **se retrouvent dans les réalisations ci-dessous** — rien d'exotique dont votre équipe aurait à hériter.

Python PostgreSQL TypeScript React GitHub Docker

Voir l'ensemble des technologies

Parcourir par catégorie

INGÉNIERIE LOGICIELLE

Ingénierie backend
Ingénierie frontend
Développement full-stack
API et intégration
Développement web
Systèmes de design et interfaces
Conception UX/UI

Tests et assurance qualité
Optimisation des performances
Python
Internationalisation

DONNÉES ET BASES DE DONNÉES

Bases de données
PostgreSQL
SQL
Ingénierie des données
Pipelines de données (ETL/ELT)
Analyse de données
Gouvernance des données
Entrepôts de données
GIS et données géospatiales

CLOUD, INFRASTRUCTURE ET DEVOPS

Cloud
Conteneurs (Docker/Kubernetes)
DevOps
Automatisation et CI/CD
Infrastructure
Linux et serveurs
Administration système
Réseaux et VPN
Supervision et observabilité
Fiabilité et sauvegardes
Sécurité
Architecture de plateformes
Migrations et modernisation

CONSEIL ET DIRECTION

Direction technique
Encadrement d'équipes
Mentorat et accompagnement
Méthodes agiles et Scrum
Gestion de projets
Produit et besoins métier
Parties prenantes et reporting
Documentation
Architecture de solutions
Marque et marketing

Tout ce qui suit **a été réalisé** — pas au stade de la proposition ou de la planification.

Chaque entrée décrit ce que la réalisation a changé et renvoie aux prestations qu'elle illustre.

Ce que nous proposons Nos tarifs →

{ Ingénierie logicielle }

Architecture de plateformes et de solutions

Décider comment construire l'ensemble

Définir d'abord le système sur le papier pour que ses éléments s'articulent et que son coût soit connu avant d'écrire du code.

Concurrence stricte de Swift 6 sur C

Adopté le mode complet de concurrence stricte de Swift 6 sans acteurs, en reliant une boucle d'événements C bloquante à l'acteur principal avec un producteur, un consommateur et un ordre unique, après avoir établi qu'une tâche par événement perdait l'ordre nécessaire à l'interface.

Un pipeline d'événements borné pour une application macOS

Borné les tampons d'événements et la journalisation après un incident macOS enregistré : croissance de 41 MB par seconde et 111 GB de pages compressées sur une machine de 36 GB ; une exécution complète des tests consignée a réduit les journaux du cœur de 610 996 à 1 411 lignes.

Client natif macOS en Swift 6

Développé un client de messagerie natif macOS en Swift 6 et SwiftUI — 11 141 lignes dans 53 fichiers — sur l'interface C d'un cœur Rust lié comme archive statique depuis une révision épinglée.

Encodeur QR intégré en 522 lignes

Intégré dans le dépôt un encodeur QR — Reed-Solomon sur GF(256), disposition fixe des modules et huit masques — en 522 lignes plutôt que d'ajouter une troisième dépendance d'exécution, puis relu la matrice finale avec un décodeur écrit indépendamment.

Générateur de CV piloté par base de données

Conçu en Python un générateur de CV, références, portfolio et lettres de motivation piloté par base de données — 41 modules, 10 580 lignes — produisant six formats à partir d'une source SQLite de 23 tables assemblée par un pipeline idempotent de 19 étapes.

Services publics sur un petit serveur

Hébergé les services publics de l'entreprise sur un hôte monocœur de 512 MB — forge git, serveur web pour sept domaines, Tor, deux serveurs de protocoles alternatifs et blocage des intrusions — avec 464 MB de mémoire utilisable comme contrainte de conception ; l'automatisation des sauvegardes constitue une mise en œuvre distincte.

Infrastructure centrale construite de zéro

Parmi les premières personnes recrutées, conçu et construit de zéro toute l'infrastructure centrale et les processus associés d'une startup SaaS d'énergie verte, posant les bases d'une croissance rapide.

Infrastructure durable, encore en service

Planifié et mis en œuvre de nouvelles fonctionnalités d'infrastructure pour des systèmes internes et externes, avec des solutions assez durables pour fonctionner encore des années plus tard avec peu de changements.

Choix techniques fondateurs de la plateforme

Défini les choix fondateurs de la plateforme dans les semaines suivant l'ouverture du dépôt en novembre 2025 — architecture en couches, accès aux données centré sur la base et zéro avertissement — toujours en vigueur neuf mois plus tard.

348 tables dans 34 schémas Postgres

Modélisé le domaine maritime dans 348 tables normalisées réparties dans 34 schémas PostgreSQL — professionnels, entreprises, navires, emplois, avis et autres entités — avec des référentiels SMALLINT et des clés UUID v7.

Couche PostgreSQL fondée sur les fonctions

Conçu une couche de données fondée sur les fonctions PostgreSQL — 1 275 fonctions stockées réparties dans 34 schémas — pour que chaque lecture et écriture passe par une fonction dont les droits sont contrôlables, plutôt que par une table.

Migration vers Huma v2 avec parité de 100 %

Migré l'API HTTP de Fiber vers Huma v2 — 649 chemins et 760 opérations — puis atteint et maintenu une correspondance de 100 % entre les routes enregistrées par le serveur et la description OpenAPI publiée.

Changement de contexte d'organisation

Conçu un système de changement de contexte d'organisation avec localStorage côté client et réplication dans un cookie côté serveur, permettant d'agir au nom des organisations administrées tout en appliquant le moindre privilège.

JSON transmis directement de Postgres à l'API

Conçu une architecture de transmission directe du JSON où les fonctions PostgreSQL renvoient des documents complets, transmis tels quels par l'API Go, supprimant la désérialisation intermédiaire et découplant le frontend des changements de schéma.

Une plateforme maritime aux frontières système claires

Conçu une plateforme maritime en couches séparant un frontend PWA Next.js, une API Go (Huma/Fiber) et une couche de fonctions PostgreSQL, avec toute la logique métier dans la base de données.

Infrastructure simplifiée au-delà d'AWS

Refondu l'infrastructure d'un client international en supprimant les services inutilisés, en ajustant la capacité et en déplaçant les charges de travail hors d'AWS. La fondatrice fait état de coûts cloud divisés par 10, sans perte de productivité.

Un produit cartographique pris en charge de bout en bout

Conçu l'architecture, développé et maintenu l'infrastructure, le traitement des données et une application cartographique pendant 2 ans, en couvrant la livraison logicielle et la maintenance d'exploitation.

Processus repensés : 8 000 heures gagnées

Refondu les processus internes et économisé 8 000 heures en améliorant l'architecture logicielle, les systèmes et la planification.

Application GIS : revenus multipliés par 10

Dirigé le développement logiciel d'une application cartographique GIS, multipliant le chiffre d'affaires par 10 et faisant du produit un actif de données central.

Un plan d'infrastructure pour 14 départements

Conçu un cadre d'infrastructure complet pour un institut de recherche regroupant 14 départements, avec des modules flexibles, des pipelines de données unifiés et des stratégies d'accompagnement structurées pour une adoption durable.

{ Ingénierie logicielle }

Développement backend et API

Le moteur de votre application ou de votre site

La partie invisible pour les utilisateurs : où résident les données, quelles règles s'appliquent et comment les autres systèmes peuvent y accéder.

Moitié jusque-là inutilisée du cœur exploitée

Exploité la moitié du cœur de messagerie jusque-là inutilisée — transfert des sauvegardes, messages éphémères, modification et renvoi de messages, invitations vérifiées, proxys et politique de chiffrement — en exécutant chaque test contre la vraie bibliothèque, sans simulations.

Parseur vérifiant chaque appel FFI

Écrit un parseur lisant le véritable en-tête C de 7 308 lignes et vérifiant chaque site d'appel, chaque constante d'énumération et le caractère final de toute classe propriétaire de pointeurs, après qu'un en-tête de substitution manuscrit a laissé compiler et lier des appels à trois fonctions supprimées, puis provoqué un plantage.

Tous les textes déplacés dans le contenu

Déplacé chaque texte destiné aux utilisateurs hors de Python vers une arborescence de 874 fichiers en trois langues, après avoir trouvé des traductions inutilisées dont l'inactivité était invisible et un contrôle n'évaluant silencieusement qu'un tiers des réalisations.

Chaque règle du linter traitée comme une erreur

Activé comme erreurs toutes les règles du linter Python, corrigé 1 815 constats pour atteindre zéro et documenté la justification de chaque rare exemption, dont deux soutenues par un contrôle plutôt que par un commentaire.

981 tests et 92 % de couverture

Vérifié le générateur avec 981 cas de test et un seuil de couverture des branches de 92 %, en traitant les avertissements comme des échecs, et vérifié l'idempotence en exécutant deux fois tout le pipeline depuis un fichier vide, sans changement autorisé au second passage.

Espaces de recrutement et carrière des marins

Construit les espaces de recrutement et de carrière des marins ; l'instantané d'implémentation consigné comprend 151 fonctions stockées dans 48 tables pour postes, candidatures, certificats, progression de grade et temps de navigation.

Couche sociale : publications, fil et groupes

Conçu la couche sociale de la plateforme — publications, fil, groupes, mentions, graphe d'abonnements et récapitulatif d'occasions — sur la même couche de données fondée sur les fonctions que le reste du produit.

Portefeuille de fidélité, ligues et badges

Conçu le système de fidélité et de réputation — 67 fonctions sur un registre de 31 tables, avec ligues, badges et boutique d'échange — en verrouillant la ligne du solde pour fermer la fenêtre de double dépense.

Limitation de débit et protection contre les abus

Conçu des protections contre les abus qui refusent l'accès en cas de défaillance — 22 limiteurs de débit Redis, Cloudflare Turnstile, idempotence des requêtes et verrouillage de l'origine — pour rejeter robots et inondations de requêtes sans faire confiance aux appelants.

Surveillance des erreurs et traçage internes

Conçu en interne la surveillance des erreurs et le traçage OpenTelemetry plutôt que de les acheter — assainissement des charges utiles, détection des pics et régressions, symbolication et signal de vie synthétique — accessibles dans 11 vues d'exploitation.

Traitements lents confiés à une file River

Sorti les traitements lents du chemin des requêtes pour les confier à une file de tâches River — 15 modules de traitement, 8 tâches planifiées et 20 tâches pg_cron — afin de répondre aux requêtes pendant la poursuite du travail en arrière-plan.

Droits d'accès Stripe et achats intégrés

Conçu la couche de paiement et de droits d'accès — Stripe et achats intégrés Apple et Google — en contrôlant l'accès à l'annuaire, à la recherche et à l'export au moyen d'un modèle de 11 tables vérifié côté serveur.

Suite de tests Go, SQL et navigateur

Conçu une suite de tests automatisés en couches — 981 tests Go, 543 spécifications frontend et navigateur, 494 tests de comportement SQL — avec tests par mutation, tests fondés sur les propriétés et contrôle obligatoire de l'accessibilité.

Applications Django pour IPTV

Développé et maintenu des applications web Django pour la plateforme IPTV, en livrant de nouvelles fonctionnalités et en améliorant les performances et la stabilité.

Application durcie avec une CSP à nonces

Durci l'application avec une CSP fondée sur des nonces, HSTS, des cookies SameSite, des rôles de base de données au moindre privilège et une nouvelle vérification des droits côté serveur.

Fusion profonde pilotée par un catalogue

Mis en œuvre une fusion profonde des préférences JSON stockées, pilotée par un catalogue, pour éviter les erreurs dues aux clés absentes lors de l'évolution du schéma.

E-mails avec basculement et journal de livraison

Conçu l'envoi d'e-mails comme une capacité de plateforme — trois fournisseurs avec basculement, webhooks de livraison, journaux d'envoi et de livraison, modèles et campagnes — avec un contrôle au démarrage qui refuse de lancer le service sans fournisseur.

Contrat d'API généré à partir de la base

Généré le contrat d'API à partir de la base de données — OpenAPI, client TypeScript typé de 44 076 lignes, 61 gestionnaires de simulation et limites appliquées par l'interface — avec un contrôle à chaque étape qui échoue en cas de divergence.

Système de reporting d'analyse de données

Développé un système de reporting d'analyse de données, augmentant de 400 % le chiffre d'affaires logiciel trimestriel grâce à des rapports PDF produits en Python.

Plus de 30 projets GIS, du code à l'assistance

Dirigé le développement, le déploiement et l'assistance de plus de 30 projets GIS, avec PostgreSQL, Bash, Python, JavaScript, GDAL, ArcGIS, PostGIS et Mapbox.

Développement cartographique GIS full-stack

Dirigé le développement full-stack d'une application cartographique GIS, avec PostgreSQL, Mapbox, ReactJS et NodeJS, pour livrer une solution intégrée.

8 projets Power BI et leurs manuels

Livré 8 projets Power BI accompagnés de manuels complets, intégrant les outils Microsoft Power BI à une API NodeJS et à Python FastAPI pour l'analyse et la visualisation des données.

6 pipelines ETL/ELT conçus et exploités

Conçu, mis en œuvre et administré 6 pipelines ETL/ELT avec Google BigQuery, MSSQL, PostgreSQL, des scripts Shell, PL/pgSQL et Transact-SQL, en intégrant les données pour leur traitement efficace par une API Python.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Concurrence stricte de Swift 6 sur C · Encodeur QR intégré en 522 lignes ·

Générateur de CV piloté par base de données · Choix techniques fondateurs de la plateforme ·

Couche PostgreSQL fondée sur les fonctions · Migration vers Huma v2 avec parité de 100 % ·

Changement de contexte d'organisation · JSON transmis directement de Postgres à l'API ·

Une plateforme maritime aux frontières système claires

{ Ingénierie logicielle }

Développement frontend

Les écrans que les gens utilisent

Des interfaces rapides, utilisables sur téléphone et cohérentes dans tous les navigateurs de vos clients.

Système de design macOS de 48 tokens

Conçu un système de design de 48 tokens sur le matériau de verre natif de la plateforme et écrit le linter refusant les nombres arbitraires, tailles de police codées en dur et animations ignorant la préférence de réduction des mouvements.

17 icônes nommées pour les lecteurs d'écran

Nommé chaque commande composée uniquement d'une icône pour les lecteurs d'écran après avoir trouvé le bouton d'envoi annoncé comme « flèche vers le haut dans un cercle, bouton », puis écrit le linter exigeant un libellé à huit lignes au plus de l'icône.

Contrôle qualité : 10 242 lignes vérifiées

Soumis 10 242 lignes de JavaScript des contrôles qualité à un formateur et à un linter, après avoir établi qu'il s'agissait du plus gros ensemble de code du dépôt et du seul que rien ne contrôlait, en corrigeant 13 constats sans en masquer aucun.

Styles allégés : de 87 KB à 31 KB

Réduit le bundle de styles de 87 KB à 31 KB en retirant une police base64, et supprimé 756 KB répartis dans 22 fichiers publiés à chaque déploiement sans être référencés nulle part.

637 lignes de styles d'impression

Conçu une feuille de style d'impression de 637 lignes tenant compte de quatre comportements documentés de moteurs de rendu, notamment des titres intégrés qui s'imprimaient en blanc sur blanc quand le navigateur préférait le thème sombre.

Réduction des mouvements réellement appliquée

Rendu effective la préférence de réduction des animations après avoir trouvé onze sélecteurs qui continuaient à animer : une règle universelle supprimant les transitions perd en spécificité face à toute règle portant une classe.

Une couche entière manquait au thème clair

Découvert qu'une couche de superposition pleine page manquait au thème clair depuis sa création, en vérifiant que les deux thèmes dessinent chaque surface composite avec le même nombre de couches.

Système limité à 28 couleurs

Limité le système de couleurs à 28 couleurs documentées, avec un linter refusant toute valeur utilisée mais non documentée, documentée mais inutilisée, mal mesurée, réécrite sous forme littérale ou située à moins de 0,02 de distance perceptuelle d'une couleur existante.

Les défauts d'accessibilité ignorés par votre outil

Détecté et corrigé 11 défauts d'accessibilité que le navigateur jugeait sains : huit liens de pied de page encore accessibles par tabulation derrière pointer-events, une chronologie de défilement masquant le colophon sur trois pages et un moteur n'exécutant que 70 de ses 105 règles.

Contrôles d’accessibilité sur 32 pages dans les deux thèmes

Mis en place des contrôles WCAG 2.2 niveau AA sur 32 pages représentatives dans les deux thèmes, avec un dossier de conformité, un critère AAA vérifié et le résultat partiel du critère 1.4.13 explicitement indiqué.

27 tailles remplacées par une échelle de 6

Remplacé 27 tailles de police rendues, dont certaines voisines n'étaient séparées que de 0,6 %, par une échelle de six tailles en tierce majeure, puis écrit le linter qui refuse la vingt-huitième.

Des vérifications qui ne prennent plus une demi-heure

Réduit les vérifications qualité du site pilotées par navigateur de 1 636 à 615 secondes, en ordonnançant les contrôles du plus long au plus court dans un pool de quatre processus, après avoir mesuré un coût de 320 secondes en ordre alphabétique contre 224.

Site Hugo multilingue en 3 langues

Conçu un site statique multilingue Hugo avec 108 fichiers de modèles, dont 53 partiels, publiant chaque page sous quatre représentations à partir d'une seule arborescence de contenu en trois langues, puis sous sept sous-domaines spécialisés issus de la même arborescence.

Back-office d'exploitation de la plateforme

Conçu le back-office d'exploitation — environ 40 routes d'administration et 128 composants pour les revendications de propriété, la modération, les indicateurs de fonctionnalités, le cache et le diagnostic — afin d'exploiter la plateforme sans accès à la base de données.

Site Hugo maintenu et interface produit améliorée

Maintenu et amélioré le site statique Hugo existant tout en contribuant à des améliorations UI/UX du produit principal de gestion d'actifs.

Système i18n anglais/danois

Mis en place un système d'internationalisation anglais/danois comportant 12 027 messages par langue dans 458 fichiers d'espaces de noms, avec vocabulaire contrôlé par linting et limite de 400 lignes par fichier.

Conception UI et UX du produit de bout en bout

Conçu l'UI et l'UX du produit de bout en bout — grilles d'annuaire, doubles vues cartes/tableau, validateurs d'exigences en temps réel et navigation par fil d'Ariane.

Système de design iOS 26 'Liquid Glass'

Conçu un système de design iOS 26 'Liquid Glass' et un inventaire de composants de référence, maintenu par des règles de linting pour prévenir les divergences d'interface.

Prise en charge complète des Progressive Web Apps

Livré une prise en charge complète des Progressive Web Apps — installation et fonctionnement hors ligne — avec mise en cache à l'exécution par Workbox via next-pwa.

Frontend Next.js 16 : SSR, SSG et CSR

Développé le frontend Next.js 16 en choisissant explicitement les stratégies SSR, SSG et CSR, avec une fabrique réutilisable de pages serveur préchargées supprimant la cascade de requêtes N+1 sur chaque page authentifiée migrée.

30 sites web, de Photoshop au HTML

Conçu 30 sites web, en réalisant des solutions originales et flexibles à partir de maquettes Photoshop converties en HTML.

100 applications web avec HTML/HTML5

Développé 100 applications web avec HTML/HTML5, CSS/SCSS, Django, WordPress et Joomla, pour répondre à différents besoins de présence en ligne.

Contribution à la conception d'interfaces

Contribué à la conception des interfaces utilisateurs pour les rendre intuitives, agréables à regarder et faciles à utiliser.

200 améliorations UI/UX présentées

Présenté 200 améliorations UI/UX pour l'application cartographique GIS, dont les nouvelles fonctionnalités ont contribué à multiplier le chiffre d'affaires par 10.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Un pipeline d'événements borné pour une application macOS · Client natif macOS en Swift 6 · Choix techniques fondateurs de la plateforme · Changement de contexte d'organisation · Une plateforme maritime aux frontières système claires · Développement cartographique GIS full-stack

{ Ingénierie logicielle }

Développement de produits full-stack

Un produit qui fonctionne, de bout en bout

Une seule équipe transforme une idée en un produit auquel vos clients peuvent se connecter, sans le faire passer entre trois prestataires.

Revendications de propriété et arbitrage

Conçu de bout en bout les revendications de propriété d'entreprise : un utilisateur revendique une entreprise, un administrateur statue, puis l'approbation réécrit le graphe d'autorisation déterminant qui peut modifier quoi.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Client natif macOS en Swift 6 · Générateur de CV piloté par base de données · Site Hugo multilingue en 3 langues · Espaces de recrutement et carrière des marins · Couche sociale : publications, fil et groupes · Back-office d'exploitation de la plateforme · Applications Django pour IPTV · Choix techniques fondateurs de la plateforme · Frontend Next.js 16 : SSR, SSG et CSR · Une plateforme maritime aux frontières système claires · Un produit cartographique pris en charge de bout en bout · 100 applications web avec HTML/HTML5 · Plus de 30 projets GIS, du code à l'assistance · Développement cartographique GIS full-stack · Application GIS : revenus multipliés par 10

{ Ingénierie logicielle }

Développement de sites web et CMS

Un site que vous pouvez mettre à jour

Un site rapide, facile à trouver et modifiable par votre équipe, sans facture à chaque changement de texte.

Miroir Tor onion à une adresse lisible

Publié un miroir Tor onion du site à une adresse lisible commençant par engineer, avec une clé générée hors de l'hôte et jamais présente dans ce dépôt ni sur un portable, en conservant ce quatrième point d'entrée comme le seul des quatre à ne jamais être compté.

Identité visuelle créée à partir de 2 dessins

Créé les icônes et favicons de l'entreprise à partir de deux dessins faits à la main, chaque ressource dérivée étant régénérée par script, avec un contrôle échouant si un fichier dérivé a été commité avant sa source.

106 contrôles sur 106 du profil PDF/UA-1 réussis

Consigné la validation PDF/UA-1 de neuf documents avec 106 vérifications sur 106 réussies dans le profil d'accessibilité configuré, tandis que la variante d'archivage échouait sur une règle parmi 146.

Dates identiques du sitemap corrigées

Corrigé un sitemap dont 172 des 176 URL partageaient la même date de modification, en utilisant l'historique git après avoir établi que l'export réécrivait chaque fichier à chaque exécution.

963 blocs de métadonnées remplacés par un graphe

Remplacé 963 blocs anonymes de données structurées répétant l'entreprise 1 671 fois par un graphe lié unique de 16 types, fondé sur des identifiants d'origine stables.

Miroirs du site sur Gemini et Gopher

Construit les sorties Gemini et Gopher depuis la même source web ; la réalisation consigne 777 documents par format à ce stade et zéro octet HTML modifié lors de leur ajout.

495 pages générées par un export SQLite

Généré 495 pages de réalisations en trois langues à partir d'un export SQLite en lecture seule, avec des adresses définies comme données pour qu'une correction de phrase ne déplace plus la page et ne casse plus son lien.

Votre code sur votre propre serveur

Déployé la forge git de l'entreprise avec Soft Serve, privée par défaut, sans interface web et avec un port SSH lié à l'interface de bouclage derrière un hôte relais, puis fait de sa page d'accueil un artefact de compilation du site principal plutôt qu'une copie maintenue à la main.

Développement web, SEO et contenu réunis

Amélioré les résultats web des clients en associant développement et conception de sites sur mesure, SEO, stratégie de contenu et rédaction.

40 sites administrés sur Apache et Nginx

Administré 40 sites web sur des serveurs d'hébergement Ubuntu Linux avec Apache et Nginx, en veillant à leur disponibilité et à leurs performances.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Styles allégés : de 87 KB à 31 KB · 637 lignes de styles d'impression · Réduction des mouvements réellement appliquée · Les défauts d'accessibilité ignorés par votre outil · Contrôles d'accessibilité sur 32 pages dans les deux thèmes · Site Hugo multilingue en 3 langues · Site Hugo maintenu et interface produit améliorée · Applications Django pour IPTV · Prise en charge complète des Progressive Web Apps · Frontend Next.js 16 : SSR, SSG et CSR · 30 sites web, de Photoshop au HTML · 100 applications web avec HTML/HTML5

{ Ingénierie logicielle }

Préparation aux agents d'IA

Être visible quand vos clients interrogent une IA

Un site ou une application que les assistants d'IA de vos clients peuvent trouver, lire et utiliser : une capacité vérifiée à chaque livraison, plutôt qu'une promesse ponctuelle.

Développement piloté par 51 initiatives écrites

Piloté le développement du site à travers 51 initiatives écrites dans 54 documents de planification, chacun conservant ses points ouverts, ses mesures et les décisions écartées, dont six des huit constats de préparation aux agents refusés avec justification consignée.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Dates identiques du sitemap corrigées · 963 blocs de métadonnées remplacés par un graphe · Contrôles d'accessibilité sur 32 pages dans les deux thèmes · Site Hugo multilingue en 3 langues

{ Ingénierie logicielle }

Outils pour agents d'IA et serveurs MCP

Permettre à l'IA d'utiliser vos logiciels existants

Vos logiciels existants exposés sous forme d'outils qu'un agent d'IA peut appeler, sans réécrire ce qui fonctionne déjà.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Back-office d'exploitation de la plateforme · Traitements lents confiés à une file River · Contrat d'API généré à partir de la base · Migration vers Huma v2 avec parité de 100 % · JSON transmis directement de Postgres à l'API

{ Ingénierie logicielle }

Solutions GIS et géospatiales

Des cartes qui répondent aux questions

Tout ce qui se situe quelque part — navires, véhicules, parcelles, capteurs — sur une carte où chercher, mesurer et partager.

30 projets GIS menés jusqu'à la livraison

Piloté l'exécution de 30 projets GIS réussis, avec une responsabilité de direction dans la livraison de solutions nouvelles dans ce domaine.

Prévisions et investissements optimisés

Optimisé les prévisions et les stratégies d'investissement de 11 gestionnaires de réseaux électriques en améliorant leur efficacité opérationnelle grâce à des solutions GIS fondées sur les données.

Livraison de 20 pipelines GIS automatisée

Automatisé la livraison de 20 pipelines de données GIS et processus ETL de données applicatives, en simplifiant l'automatisation de l'infrastructure et le reporting.

Déploiement SaaS GIS automatisé : GitHub Actions

Automatisé le déploiement d'une application SaaS GIS, le traitement des données et le système de reporting avec GitHub Actions CI/CD, Python, Bash et SQL.

100 entrepôts PostgreSQL, MS SQL et BigQuery

Conçu l'architecture, créé et géré 100 bases d'entrepôts de données PostgreSQL, MS SQL et Google BigQuery, principalement consacrées aux données GIS et aux séries temporelles, en optimisant leurs performances et leur capacité de montée en charge.

500 rapports de données électriques et GIS

Publié 500 rapports d'analyse de données électriques et GIS, fondés sur des recherches approfondies et la résolution de problèmes pour fiabiliser les analyses de données massives géographiques et temporelles.

1 000 problèmes de données géographiques résolus

Résolu 1 000 problèmes de données géographiques et de séries temporelles avec GDAL, ArcGIS, PostGIS, Mapbox, QGIS, SQL (PL/pgSQL, Transact-SQL) et Bash, pour un traitement de qualité des données massives.

Cartes GIS 10 fois plus rapides sur PostgreSQL

Accéléré de 10 fois une application cartographique en migrant stratégiquement de MSSQL vers PostgreSQL, basculement des données compris, et en optimisant les traitements et la sécurité des données.

Pipeline géographique accéléré de 50 fois

Consigné une accélération de 50 fois d'un pipeline géographique dans les travaux de la fondatrice chez Utiligize, grâce au SQL et à la modélisation sur PostgreSQL, MS SQL et Google Cloud BigQuery.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Un produit cartographique pris en charge de bout en bout ·

Plus de 30 projets GIS, du code à l'assistance · Développement cartographique GIS full-stack ·

Application GIS : revenus multipliés par 10

{ Ingénierie logicielle }

Conception UI/UX et systèmes de design

Rendre l'utilisation évidente

Une organisation des écrans, des textes et des parcours qui permet aux utilisateurs d'accomplir leur tâche sans appeler l'assistance.

Erreurs réécrites sans blâmer les utilisateurs

Réécrit les messages d'erreur de l'application selon une charte de ton après qu'un refus de connexion reprochait une faute de frappe à l'utilisateur alors que le fournisseur exigeait un mot de passe d'application, avec un test nommant ce fournisseur.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Identité visuelle créée à partir de 2 dessins · Système de design macOS de 48 tokens · 17 icônes nommées pour les lecteurs d'écran · Client natif macOS en Swift 6 · 106 contrôles sur 106 du profil PDF/UA-1 réussis · 637 lignes de styles d'impression · Réduction des mouvements réellement appliquée · Une couche entière manquait au thème clair · Système limité à 28 couleurs · Les défauts d'accessibilité ignorés par votre outil · Contrôles d’accessibilité sur 32 pages dans les deux thèmes · 27 tailles remplacées par une échelle de 6 · Développement web, SEO et contenu réunis · Site Hugo maintenu et interface produit améliorée · Conception UI et UX du produit de bout en bout · Système de design iOS 26 'Liquid Glass' · 30 sites web, de Photoshop au HTML · Contribution à la conception d'interfaces · 200 améliorations UI/UX présentées

{ Ingénierie logicielle }

Internationalisation et localisation

Fonctionner dans plusieurs langues

Un logiciel qui se lit naturellement sur chaque marché où vous vendez, y compris les dates, les devises et le classement des noms.

Contrôle des chiffres entre langues

Conçu un contrôle multilingue échouant quand une traduction omet un chiffre présent en anglais ou utilise une notation étrangère à sa langue, révélant deux descriptions danoises privées d'une mesure et seize passages ukrainiens utilisant les guillemets anglais.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Tous les textes déplacés dans le contenu · Générateur de CV piloté par base de données · Miroirs du site sur Gemini et Gopher · 495 pages générées par un export SQLite · Contrôles d'accessibilité sur 32 pages dans les deux thèmes · Site Hugo multilingue en 3 langues · Système i18n anglais/danois

{ Ingénierie logicielle }

Développement d'agents d'IA sur mesure

Un assistant d'IA qui connaît votre métier

Un agent connecté à vos systèmes, testé sur des cas rédigés par vos experts, supervisé par votre équipe et indépendant d'un modèle unique.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Six collecteurs indiquent les entrées ignorées · Tests des contrôleurs eux-mêmes · Contrôle avant commit avec 22 linters · Surveillance des erreurs et traçage internes · Moteur de 268 contrôles pour le dépôt

{ Données et bases de données }

Conception et modélisation de bases de données

Un endroit bien pensé pour vos données

Une structure fidèle au fonctionnement de l'entreprise, qui permet encore de poser demain les questions dont vous aurez besoin.

Analytique interne dans PostgreSQL

Conçu une analytique produit interne dans PostgreSQL — 47 fonctions sur des tables d'événements partitionnées avec purge automatique — pseudonymisée par un sel renouvelé et soumise au consentement du visiteur.

Des clés chronologiques pour un modèle de données qui grandit

Adopté des identifiants UUID v7 ordonnés dans le temps (PostgreSQL 18) comme clés d'entités pour réduire la fragmentation des index B-tree et accélérer les requêtes.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Générateur de CV piloté par base de données · Espaces de recrutement et carrière des marins · Portefeuille de fidélité, ligues et badges · Traitements lents confiés à une file River · Droits d'accès Stripe et achats intégrés · 348 tables dans 34 schémas Postgres · Fusion profonde pilotée par un catalogue · Couche PostgreSQL fondée sur les fonctions · JSON transmis directement de Postgres à l'API · Une plateforme maritime aux frontières système claires · 100 entrepôts PostgreSQL, MS SQL et BigQuery · 6 pipelines ETL/ELT conçus et exploités

{ Données et bases de données }

Administration de bases de données (DBA)

Prendre soin de la base de données

Une personne responsable du socle de toute l'entreprise : il fonctionne, ses données sont sauvegardées et il ne se dégrade pas en silence.

1 022 migrations contrôlées contre les écarts

Maintenu la cohérence du schéma au fil de 1 022 migrations avec un contrôle CI construisant la base de deux façons — installation neuve, puis installation suivie de toutes les migrations — et échouant en cas de différence.

Sauvegardes et reprise sous forme de code

Configuré la conservation des sauvegardes sous forme d'Infrastructure as Code, puis évalué les possibilités de récupération et documenté la restauration, en explicitant les lacunes restantes avant qu'un incident ne les révèle.

600 pipelines ETL/ELT PL/pgSQL conçus

Conçu 600 pipelines ETL/ELT en PL/pgSQL pour simplifier des traitements de données complexes dans plusieurs environnements PostgreSQL de développement et de production.

Traitement automatisé : Python et PL/pgSQL

Automatisé les traitements de données avec des scripts Shell, PL/pgSQL, Python et Transact-SQL pour accroître la productivité et l'efficacité.

100 sauvegardes critiques automatisées : Barman

Automatisé 100 sauvegardes critiques avec Barman, Google Cloud, Bash et Python pour préserver l'intégrité des données dans les bases.

PostgreSQL accéléré de 10 fois

Accéléré PostgreSQL de 10 fois grâce à l'indexation, au partitionnement et à l'optimisation des requêtes sur les données des utilisateurs, des organisations clientes, les données géospatiales et les séries temporelles électriques.

1 000 règles RBAC pour sécuriser Linux

Renforcé la sécurité des données en mettant en place 1 000 règles RBAC pour les développeurs, les instances applicatives, PostgreSQL, MS SQL et d'autres serveurs Linux afin de prévenir les accès non autorisés, avec une documentation et une automatisation Ansible.

10 serveurs PostgreSQL et MS SQL sur Ubuntu

Conçu, déployé et maintenu 10 serveurs PostgreSQL et MS SQL sur des VPS Ubuntu Linux, en veillant à leurs performances et à leur fiabilité.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

100 entrepôts PostgreSQL, MS SQL et BigQuery

{ Données et bases de données }

Optimisation des performances de bases de données

Accélérer ce qui ralentit tout

Identifier les requêtes et les index manquants dans un système alourdi au fil des ans, puis corriger ce qui compte.

30 jours de données électriques traités en 180 ms

Conçu des variantes d'agrégation horaire de la consommation électrique en Python, SQL et Bash + Jq ; le benchmark Python a traité un jeu de données JSONL de 30 jours en 180 ms.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Des clés chronologiques pour un modèle de données qui grandit · PostgreSQL accéléré de 10 fois · 100 entrepôts PostgreSQL, MS SQL et BigQuery · Cartes GIS 10 fois plus rapides sur PostgreSQL · Pipeline géographique accéléré de 50 fois

{ Données et bases de données }

Migration et modernisation de bases de données

Quitter l'ancien système

Transférer les données d'un système vieillissant vers une solution maintenue, tout en poursuivant l'activité pendant la migration.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Tous les textes déplacés dans le contenu · 1 022 migrations contrôlées contre les écarts · Cartes GIS 10 fois plus rapides sur PostgreSQL

{ Données et bases de données }

Développement de pipelines de données (ETL/ELT)

Acheminer vos données de A à B, automatiquement

Des chiffres disponibles là où il faut, à la fréquence prévue, sans recopier un tableur à la fin du mois.

4 000 h gagnées avec GitHub et GitLab CI/CD

Simplifié les processus d'analyse de données et de développement logiciel et économisé 4 000 heures en introduisant des pratiques CI/CD avec GitHub, GitLab, Bash et Python.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

495 pages générées par un export SQLite · Analytique interne dans PostgreSQL · Un produit cartographique pris en charge de bout en bout · 600 pipelines ETL/ELT PL/pgSQL conçus · Traitement automatisé : Python et PL/pgSQL · Plus de 30 projets GIS, du code à l'assistance · Livraison de 20 pipelines GIS automatisée · Déploiement SaaS GIS automatisé : GitHub Actions · 6 pipelines ETL/ELT conçus et exploités · 1 000 problèmes de données géographiques résolus · Pipeline géographique accéléré de 50 fois · 30 jours de données électriques traités en 180 ms · Un plan d'infrastructure pour 14 départements

{ Données et bases de données }

Conception d'entrepôts de données

Un endroit où tous les chiffres concordent

Réunir les chiffres des ventes, des opérations et des finances pour que deux services ne présentent plus deux réponses à la même réunion.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

100 entrepôts PostgreSQL, MS SQL et BigQuery · 6 pipelines ETL/ELT conçus et exploités · Pipeline géographique accéléré de 50 fois

{ Données et bases de données }

Analyse de données et tableaux de bord BI

Comprendre ce qui se passe

Des tableaux de bord et des rapports présentant les quelques chiffres qui déterminent une décision, actualisés automatiquement.

Gestion du temps et reporting

Conçu un système de gestion du temps et de reporting resté en production pendant des années sans modification importante.

Analyses et rapports d'avancement réguliers

Fourni au CEO des analyses et des rapports d'avancement réguliers, en traduisant les indicateurs d'ingénierie et l'état des livraisons en éléments de décision.

Premier tableau de bord d'observabilité

Développé et lancé le premier tableau de bord d'observabilité de l'entreprise, affichant en temps réel les performances du système et des visualisations de données sur le grand téléviseur des bureaux.

Une communication de marque éclairée par le public

Utilisé les études de marché et les retours du public pour orienter la communication et les prises de contact d'Engineer ApS.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Analytique interne dans PostgreSQL · Système de reporting d'analyse de données ·
4 000 h gagnées avec GitHub et GitLab CI/CD · Prévisions et investissements optimisés ·
500 rapports de données électriques et GIS · 8 projets Power BI et leurs manuels ·
30 jours de données électriques traités en 180 ms

{ Données et bases de données }

Gouvernance et qualité des données

Pouvoir se fier aux chiffres

Des règles et des contrôles qui détectent les données incorrectes dès leur arrivée, avec une réponse claire sur le responsable et l'origine de chaque chiffre.

13 adresses mesurées sans traçage

Mesuré treize adresses uniquement à partir du journal d'accès du serveur web — sans cookie, traqueur ni service tiers — en masquant le réseau du client à l'écriture de chaque ligne, puis en intégrant les résultats dans une archive permanente de 21 comptes quotidiens et 20 facettes mensuelles.

Connaître la licence de tout ce que vous livrez

Audité à chaque compilation la compatibilité des licences de 644 crates Rust, puis vérifié le déclenchement du contrôle en modifiant la licence d'une crate et en déplaçant la révision épinglée du cœur sans régénération.

Tests des contrôleurs eux-mêmes

Écrit des tests pour les contrôleurs après avoir établi qu'un contrôleur alimenté uniquement en entrées correctes peut finir par déclarer une réussite sans rien lire : introduction d'une faute pour vérifier sa détection et lecture d'une plage d'identifiants dans la base plutôt que dans une constante du test.

Règle de périmètre après une fuite de secrets

Écrit une règle de périmètre dans le dépôt après qu'une restructuration y a introduit l'inventaire, les autorisations de pare-feu et les textes d'une autre entreprise, en conservant les éléments mis en quarantaine dans le champ de la détection des secrets.

Besoins métier recueillis et analysés

Recueilli et analysé les besoins métier pour les traduire en fonctionnalités réalisables et en récits utilisateurs conformes aux exigences de gouvernance des données.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Contrôle des chiffres entre langues · Tous les textes déplacés dans le contenu ·
963 blocs de métadonnées remplacés par un graphe · Analytique interne dans PostgreSQL ·
348 tables dans 34 schémas Postgres · Fusion profonde pilotée par un catalogue ·

Couche PostgreSQL fondée sur les fonctions · 600 pipelines ETL/ELT PL/pgSQL conçus · 500 rapports de données électriques et GIS · 1 000 problèmes de données géographiques résolus · Un plan d'infrastructure pour 14 départements

{ Données et bases de données }

Préparation des données à l'IA

Des données que l'IA peut réellement utiliser

Des données accessibles aux agents d'IA, une seule fiche par client, des accès limités au nécessaire et chaque réponse reliée à sa source.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

963 blocs de métadonnées remplacés par un graphe · 495 pages générées par un export SQLite · 1 022 migrations contrôlées contre les écarts · 348 tables dans 34 schémas Postgres · 6 pipelines ETL/ELT conçus et exploités · Pipeline géographique accéléré de 50 fois

{ Cloud, infrastructure et DevOps }

Infrastructure cloud et migration

L'endroit où fonctionne votre logiciel

Des serveurs dimensionnés selon vos usages réels, migrés vers le cloud ou hors du cloud, avec une estimation préalable de la facture mensuelle.

Second serveur provisionné à partir du code

Automatisé le provisionnement d'un second serveur chez un second fournisseur cloud, en créant le pare-feu avant la machine pour qu'elle soit protégée dès sa création, et configuré les pare-feu des deux fournisseurs directement par leurs API REST pour éviter une collection tierce.

Conteneurs exploités avec Podman et Quadlet

Déployé une couche de conteneurs Podman et Quadlet sous systemd plutôt que Docker, car la publication de ports sur le réseau bridge par défaut de Docker peut contourner les règles UFW, et confié les déploiements à un utilisateur non privilégié autorisé à une seule commande fixe plutôt qu'à root.

Une infrastructure définie dans un dépôt

Conçu l'infrastructure de l'entreprise avec 19 playbooks Ansible et 34 rôles totalisant 12 065 lignes de YAML, pour faire converger un hôte en service vers un état déclaré, chaque play étant idempotent.

Déploiements Azure pris en charge de bout en bout

Pris en charge les déploiements de bout en bout sur Azure, en gérant les livraisons dans les environnements de développement, de préproduction et de production.

Infrastructure Azure sous forme de code

Provisionné l'infrastructure Azure sous forme de code avec Bicep — Container Apps, PostgreSQL Flexible Server, Front Door/WAF et réseau — dans les environnements de développement, de préproduction et de production.

20 applications Docker sur Google Cloud et AWS

Déployé et maintenu 20 applications conteneurisées avec Docker, résolu des incidents avec Podman et Kubernetes et géré des applications R sur Google Cloud et AWS.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Services publics sur un petit serveur · Infrastructure centrale construite de zéro · Infrastructure simplifiée au-delà d'AWS

{ Cloud, infrastructure et DevOps }

Infrastructure as Code

Des serveurs que vous pouvez reconstruire à partir d’une configuration versionnée

Toute la configuration décrite dans des fichiers, pour la recréer à l'identique sans qu'elle ne dépende de la mémoire d'une seule personne.

Hook CI de forge répété : 2 défauts détectés

Répété l'exécution du hook CI côté forge et trouvé deux défauts impossibles à atteindre par la seule lecture : un mécanisme de repli fournissant un argument impossible à résoudre à l'entrée du hook et une variable d'environnement git suivant le contrôle dans le checkout, faisant échouer 19 tests.

Secrets d'exploitation séparés par hôte

Séparé les quatre secrets d'exploitation par hôte après avoir établi que deux hôtes partageant un seul signal de vie produisent moins d'alertes que deux signaux distincts, et qu'une phrase secrète de sauvegarde commune peut exposer les sauvegardes des deux hôtes, même dans des dépôts distincts, si une personne l'obtient et peut lire les données chiffrées et les fichiers de clés des deux dépôts.

Exposition des bacs à sable systemd réduite

Réduit l'exposition des bacs à sable systemd de chaque unité installée par la plateforme — service de signal de vie de 9,6 UNSAFE à 1,5, forge git exposée à Internet de 8,3 EXPOSED à 1,5 — et ajouté un contrôle du parseur lors de la convergence après avoir découvert une directive mal orthographiée silencieusement ignorée dans trois modèles d'unités.

Zone DNS de 20 enregistrements en code

Réconcilié déclarativement une zone DNS de 20 enregistrements avec l'API Cloudflare, avec des points d'entrée distincts pour l'audit et l'export BIND, puis désactivé le proxy CDN après sa mise en place pour des raisons de confidentialité.

Vérification des secrets en une seconde

Ajouté un play de vérification préalable exécutant en une seconde environ le même code que la convergence, avec les secrets locaux de l'opérateur, après qu'une exécution de production partiellement appliquée s'est arrêtée à la neuvième tâche, les paramètres de swap étant déjà écrits sur l'hôte en service.

Mode de vérification Ansible fiabilisé

Fiabilisé le mode de vérification sur toute la plateforme après avoir trouvé six sondes prenant des décisions sur une valeur que l'hôte n'avait jamais fournie : le module command d'Ansible annonce une réussite sous --check tout en sautant entièrement la commande.

Supervision par absence de signal de vie

Conçu une supervision par absence de signal de vie qui n'émet un signal que si mémoire et disque sont sains, pour qu'un hôte dégradé déclenche une alerte en se taisant, et corrigé six noms de variables indiquant « free » alors que le contrôle mesurait bien « available », des valeurs d'un ordre de grandeur différent sur une machine de 464 MB.

Sauvegardes automatisées et méthode de restauration explicite

Mis en place l'automatisation des sauvegardes restic avec conservation, contrôles d'intégrité et procédure mensuelle de test de restauration ; documenté l'activation du stockage et la vérification d'une restauration réelle comme des étapes d'exploitation distinctes.

SSH durci avec 24 directives validées

Durci SSH avec 24 directives vérifiées et une validation en trois étapes — fichier candidat, configuration assemblée, puis relecture par le démon — après que cette dernière a révélé que le serveur en service remplaçait silencieusement deux des vingt-quatre directives.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Miroir Tor onion à une adresse lisible · Second serveur provisionné à partir du code ·

Votre code sur votre propre serveur · Services publics sur un petit serveur ·

Une infrastructure définie dans un dépôt · Sauvegardes et reprise sous forme de code ·

Infrastructure Azure sous forme de code · 1 000 règles RBAC pour sécuriser Linux

{ Cloud, infrastructure et DevOps }

DevOps et automatisation CI/CD

Livrer des changements sans agitation

Des changements testés, publiés et réversibles, pour livrer aux clients un mardi après-midi plutôt qu'un vendredi soir tard.

Une norme d'historique pour 4 dépôts

Appliqué à quatre dépôts une même norme d'historique — commits conventionnels, sans emojis ni mention d'attribution, imposés par un hook de message de commit — et 48 documents d'instructions régissant le travail.

Suite de tests ramenée à 8,9 secondes

Fait passer la suite de six tests dépassant soixante secondes à 135 tests réussis en 8,9 secondes, en profilant le thread principal et en supprimant les deux appels où il restait bloqué pendant 3 989 des 4 017 échantillons.

Divergences entre hooks et contrôle qualité

Découvert que les hooks de commit et le contrôle qualité exécutaient des vérifications différentes alors qu'un document les disait identiques, en comparant les deux listes dans un test : les cinq contrôles seulement lancés à la main étaient ceux qui lisaient les textes du CV.

Contrôle avant commit avec 22 linters

Conçu un contrôle obligatoire avant commit composé de 22 linters tenant sur une ligne et de cinq nécessitant un paragraphe, sans niveau d'avertissement ni suppression locale autorisée, couvrant HTML, CSS, JavaScript, Python, YAML, Markdown, shell, liens, orthographe, secrets et typographie.

Programme continu d'analyse de sécurité

Mis en place un programme continu de sécurité — analyse du code, DAST, contrôles des dépendances et des vulnérabilités, génération de SBOM, détection de secrets et actions épinglées par SHA — accompagné de 21 audits de sécurité écrits.

Moteur de 268 contrôles pour le dépôt

Conçu le moteur de garde-fous du dépôt — 268 contrôles de commit enregistrés, 277 règles de linting et 15 règles ESLint personnalisées — ainsi que 146 tests de ces garde-fous, pour que les exigences soient appliquées par la compilation et non par la seule revue.

Collaboration d'équipe automatisée

Automatisé la collaboration, la gestion des mots de passe, des tâches et du temps, et construit un système semi-automatique de présentation des projets pour accroître la productivité de l'équipe.

Zéro avertissement dans six langages

Fixé un niveau d'exigence de zéro avertissement dans six langages — Go, TypeScript, SQL, Python, Shell et Markdown — appliqué par des hooks pre-commit.

578 cibles d'automatisation go-task

Créé 578 cibles d'automatisation go-task couvrant les modes de développement natif, Docker et HTTPS, les contrôles de code, les tests, la base de données et le déploiement.

Pipelines CI/CD GitHub Actions

Conçu des pipelines GitHub Actions CI/CD avec une image frontend de production distroless et une promotion entre plusieurs environnements.

CI/CD simplifiée : 4 000 heures gagnées

Simplifié les processus CI/CD et économisé 4 000 heures grâce à l'automatisation des pipelines de développement logiciel.

SSL/TLS automatisé pour 100 applications Docker

Automatisé la création de certificats SSL/TLS pour 100 applications Docker afin de sécuriser les connexions sur des hôtes Ubuntu Linux.

Risques réduits avec Grafana et Prometheus

Réduit les risques d'exploitation en mettant en place un tableau de bord de supervision avec Grafana et Prometheus pour améliorer la fiabilité du système.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Connaître la licence de tout ce que vous livrez · Moitié jusque-là inutilisée du cœur exploitée · Parseur vérifiant chaque appel FFI · Hook CI de forge répété : 2 défauts détectés · Tests des contrôleurs eux-mêmes · Chaque règle du linter traitée comme une erreur · 981 tests et 92 % de couverture · Contrôle qualité : 10 242 lignes vérifiées · Dates identiques du sitemap corrigées · Miroirs du site sur Gemini et Gopher · Des vérifications qui ne prennent plus une demi-heure · Conteneurs exploités avec Podman et Quadlet · Votre code sur votre propre serveur · Vérification des secrets en une seconde · Mode de vérification Ansible fiabilisé · Une infrastructure définie dans un dépôt · 1 022 migrations contrôlées contre les écarts · Surveillance des erreurs et traçage internes · Suite de tests Go, SQL et navigateur · Infrastructure centrale construite de zéro · Déploiements Azure pris en charge de bout en bout · Contrat d'API généré à partir de la base · Infrastructure simplifiée au-delà d'AWS · Traitement automatisé : Python et PL/pgSQL · 4 000 h gagnées avec GitHub et GitLab CI/CD · Processus repensés : 8 000 heures gagnées · 20 applications Docker sur Google Cloud et AWS · 100 sauvegardes critiques automatisées : Barman · Livraison de 20 pipelines GIS automatisée · Déploiement SaaS GIS automatisé : GitHub Actions

{ Cloud, infrastructure et DevOps }

Conteneurisation et orchestration

Un logiciel qui fonctionne partout de la même façon

Un logiciel conditionné pour se comporter à l'identique
sur un portable, un serveur de test et en production,
et monter en charge lors des semaines chargées.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Conteneurs exploités avec Podman et Quadlet · Exposition des bacs à sable systemd réduite · Pipelines CI/CD GitHub Actions · SSL/TLS automatisé pour 100 applications Docker · 20 applications Docker sur Google Cloud et AWS

{ Cloud, infrastructure et DevOps }

Fiabilité des services et supervision

Être informé avant vos clients

Une supervision et des alertes qui signalent les incidents tant qu'ils restent mineurs, avec les chiffres qui montrent à quelle fréquence ils le restent.

Six collecteurs indiquent les entrées ignorées

Fait expliciter à six collecteurs de reporting la partie de leurs entrées jamais lue — fichiers non ouverts, sondes non installées, lignes du journal non reconnues — et ajusté les seuils sur 116 échantillons mesurés, après avoir établi qu'une collecte aveugle et un hôte sain produisaient la même page.

7 rôles de reporting de l'hôte en lecture seule

Conçu sept rôles de reporting en lecture seule décrivant un hôte en service en Markdown — faits, accès, git, métriques, trafic, sécurité et inventaire fournisseur — selon une règle interdisant d'afficher un chiffre non mesuré pendant l'exécution.

Bannissement testé à chaque exécution

Vérifié de bout en bout le blocage des intrusions à chaque exécution du durcissement, en bannissant une adresse réservée aux tests, en lisant la règle noyau obtenue et en levant le bannissement dans un bloc de nettoyage garanti, pour qu'une protection défaillante fasse échouer l'exécution au lieu de se déclarer saine.

Trois verrous sur la porte, aucun fermé

Détecté et corrigé trois protections SSH contre la force brute qui n'avaient jamais fonctionné : une prison de bannissement surveillant le port 22 alors que le démon écoutait sur 1986, une limitation de débit masquée par une règle plus large placée avant elle et une action de bannissement dont le binaire était introuvable, empêchant tout bannissement.

Continuité des flux de diffusion IPTV

Assuré la continuité de l'acheminement des flux IPTV entre fournisseurs et clients, en supervisant et maintenant le réseau de diffusion et la téléphonie IP jour et nuit.

Assistance 24/7 : environ 1 000 serveurs IPTV/OTT

Assuré une assistance infrastructure 24/7 pour une plateforme de diffusion IPTV/OTT, en administrant environ 1 000 serveurs et des systèmes appartenant aux clients pour une clientèle mondiale en Chine, aux États-Unis et en Allemagne.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

13 adresses mesurées sans traçage · Un pipeline d'événements borné pour une application macOS · Secrets d'exploitation séparés par hôte · Mode de vérification Ansible fiabilisé · Supervision par absence de signal de vie · Sauvegardes automatisées et méthode de restauration explicite · Limitation de débit et protection contre les abus · Surveillance des erreurs et traçage internes · Traitements lents confiés à une file River · Infrastructure durable, encore en service · Déploiements Azure pris en charge de bout en bout · E-mails avec basculement et journal de livraison · Un produit cartographique pris en charge de bout en bout · 40 sites administrés sur Apache et Nginx · Risques réduits avec Grafana et Prometheus · 10 serveurs PostgreSQL et MS SQL sur Ubuntu · Premier tableau de bord d'observabilité

{ Cloud, infrastructure et DevOps }

Sauvegarde et reprise après sinistre

Tout retrouver après une mauvaise journée

Des sauvegardes testées plutôt que supposées fiables, et un plan écrit pour le matin où un serveur ne redémarre pas.

30 instances VPS Ubuntu Linux administrées

Géré 30 instances VPS Ubuntu Linux, mis en place des stratégies de reprise après sinistre et adapté les configurations réseau.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Secrets d'exploitation séparés par hôte ·

Sauvegardes automatisées et méthode de restauration explicite ·

Sauvegardes et reprise sous forme de code · 100 sauvegardes critiques automatisées : Barman

{ Cloud, infrastructure et DevOps }

Administration système

Faire fonctionner l'essentiel

L'entretien quotidien des serveurs, de la messagerie, des comptes et des mises à jour, pour que les outils indispensables restent sans histoire.

Règles de pare-feu vérifiées par position

Vérifié les règles du pare-feu selon leur position plutôt que leur seule présence, en lisant la liste numérotée et la chaîne active de filtrage des paquets : une règle existante n'est pas forcément atteinte par le moindre paquet.

Gestionnaire de mots de passe commun

Intégré un système de gestion des mots de passe à l'échelle de l'entreprise pour renforcer la sécurité et simplifier le contrôle des accès.

100 serveurs Bare Bone administrés

Administré 100 serveurs Bare Bone, des réseaux physiques et des systèmes de téléphonie IP pour soutenir la croissance de l'entreprise.

1 000 routeurs Wi-Fi configurés et déployés

Configuré et déployé 1 000 routeurs Wi-Fi pour améliorer l'accès au réseau et ses performances pour les clients.

Infrastructure réseau administrée

Administré l'infrastructure réseau de plus de 1 000 serveurs, en veillant à leur déploiement, à leur sécurité et à la résolution des incidents.

8 connexions VPN WireGuard et IPsec gérées

Géré et dépanné 8 connexions VPN WireGuard et IPsec, en sécurisant les communications entre Google Cloud et les systèmes Linux.

Gestion des accès avec M365 et SSO

Prévenu des incidents de sécurité en pilotant la gestion des accès avec M365, 1Password, Red Hat SSO et OKTA SSO.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Miroir Tor onion à une adresse lisible · Six collecteurs indiquent les entrées ignorées · Hook CI de forge répété : 2 défauts détectés · Votre code sur votre propre serveur · 7 rôles de reporting de l'hôte en lecture seule · Exposition des bacs à sable systemd réduite · Zone DNS de 20 enregistrements en code · Supervision par absence de signal de vie · Sauvegardes automatisées et méthode de restauration explicite · Bannissement testé à chaque exécution · SSH durci avec 24 directives validées · Services publics sur un petit serveur · Une infrastructure définie dans un dépôt · Infrastructure centrale construite de zéro · Infrastructure durable, encore en service · Continuité des flux de diffusion IPTV · Assistance 24/7 : environ 1 000 serveurs IPTV/OTT · 40 sites administrés sur Apache et Nginx · 30 instances VPS Ubuntu Linux administrées · 1 000 règles RBAC pour sécuriser Linux · 10 serveurs PostgreSQL et MS SQL sur Ubuntu

{ Cloud, infrastructure et DevOps }

Réseaux et configuration de VPN

Relier vos bureaux et vos équipes

Du Wi-Fi, des pare-feu et des accès à distance configurés pour travailler de partout, en laissant entrer uniquement les personnes autorisées.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Second serveur provisionné à partir du code · Zone DNS de 20 enregistrements en code · Règles de pare-feu vérifiées par position · Trois verrous sur la porte, aucun fermé · Continuité des flux de diffusion IPTV · Infrastructure Azure sous forme de code · 100 serveurs Bare Bone administrés · 1 000 routeurs Wi-Fi configurés et déployés · Infrastructure réseau administrée · 8 connexions VPN WireGuard et IPsec gérées · 30 instances VPS Ubuntu Linux administrées

{ Cloud, infrastructure et DevOps }

Sécurité et gestion des accès

Décider qui peut accéder à quoi

Des comptes, mots de passe et droits attribués selon les besoins exacts de chaque poste, puis retirés le jour du départ.

Communication et collaboration améliorées

Amélioré la communication et la collaboration de l'équipe avec Slack, Mattermost, 1Password et Jira, économisant 8 000 heures de travail.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Miroir Tor onion à une adresse lisible · 13 adresses mesurées sans traçage · Connaître la licence de tout ce que vous livrez · Moitié jusque-là inutilisée du cœur exploitée · Secrets d'exploitation séparés par hôte · Règle de périmètre après une fuite de secrets · Second serveur provisionné à partir du code · Conteneurs exploités avec Podman et Quadlet · Exposition des bacs à sable systemd réduite · Zone DNS de 20 enregistrements en code · Vérification des secrets en une seconde · Règles de pare-feu vérifiées par position · Bannissement testé à chaque exécution · SSH durci avec 24 directives validées · Trois verrous sur la porte, aucun fermé · Programme continu d'analyse de sécurité · Revendications de propriété et arbitrage · Limitation de débit et protection contre les abus · Droits d'accès Stripe et achats intégrés · Gestionnaire de mots de passe commun · Collaboration d'équipe automatisée · Application durcie avec une CSP à nonces · Infrastructure Azure sous forme de code · E-mails avec basculement et journal de livraison · Changement de contexte d'organisation · SSL/TLS automatisé pour 100 applications Docker · Infrastructure réseau administrée · 8 connexions VPN WireGuard et IPsec gérées · Gestion des accès avec M365 et SSO · 1 000 règles RBAC pour sécuriser Linux

{ Cloud, infrastructure et DevOps }

Sécurité des agents d'IA

Limiter votre IA à ce qu'elle doit faire

Les accès de chaque agent recensés et limités au nécessaire, chaque action consignée hors de sa portée et chaque protection testée plutôt que supposée fiable.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Secrets d'exploitation séparés par hôte · Règle de périmètre après une fuite de secrets · Vérification des secrets en une seconde · Bannissement testé à chaque exécution · Trois verrous sur la porte, aucun fermé · Programme continu d'analyse de sécurité · Limitation de débit et protection contre les abus · Gestion des accès avec M365 et SSO · 1 000 règles RBAC pour sécuriser Linux

{ Conseil et direction }

Direction technique et conseil

Une ingénieure expérimentée à vos côtés

Quelqu'un de votre côté de la table pour apporter un second avis éclairé sur un fournisseur, un budget ou une décision d'architecture.

Documentation soumise à son propre linter

Soumis la documentation à son propre linter — budget de lignes par fichier ne pouvant que diminuer, obligation d'index et contrôle de l'existence des chemins — qui a trouvé six plans sur neuf déclarés terminés avec des cases non cochées et quatorze tâches obligatoires avant commit absentes des documents.

Projets accompagnés durant le développement

Guidé l'exécution de nombreux projets de l'entreprise en apportant une expertise pendant les phases de développement logiciel.

Croissance de l'entreprise de 4 à 14 salariés

Accompagné la croissance de l'entreprise de 4 à 14 salariés en appliquant des méthodes agiles et des pratiques efficaces de gestion de projets.

50 000 mots de documentation logicielle

Rédigé 50 000 mots de documentation logicielle complète en Markdown dans GitHub, Craft et Confluence pour conserver les connaissances et rendre le processus transparent.

Gain estimé : 4 000 h, équipe passée de 2 à 18

Estimé à 4 000 heures le temps économisé en accompagnant une équipe passée de 2 à 18 personnes, en améliorant les méthodes de travail et la coordination entre départements.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Une norme d'historique pour 4 dépôts · Suite de tests ramenée à 8,9 secondes · Concurrence stricte de Swift 6 sur C · Divergences entre hooks et contrôle qualité · Tests des contrôleurs eux-mêmes · Chaque règle du linter traitée comme une erreur · 981 tests et 92 % de couverture · Contrôle qualité : 10 242 lignes vérifiées · Développement piloté par 51 initiatives écrites · Des vérifications qui ne prennent plus une demi-heure · Moteur de 268 contrôles pour le dépôt · Suite de tests Go, SQL et navigateur · Analyses et rapports d'avancement réguliers · Choix techniques fondateurs de la plateforme · Zéro avertissement dans six langages · CI/CD simplifiée : 4 000 heures gagnées · Processus repensés : 8 000 heures gagnées · Plus de 30 projets GIS, du code à l'assistance · Développement cartographique GIS full-stack · Application GIS : revenus multipliés par 10 · Un plan d'infrastructure pour 14 départements

{ Conseil et direction }

Constitution d'équipes et mentorat

Faire grandir votre équipe

Recrutement, intégration et accompagnement pour que le savoir reste dans votre équipe plutôt que de partir avec une consultante.

Pilotage d'une équipe d'assistance informatique

Dirigé une équipe fournissant une assistance informatique, de la récupération de données et de la réparation de matériel à plus de 1 000 clients, avec une exigence de qualité.

150 h gagnées par mois sur 30 projets

Amélioré l'efficacité des projets et économisé 150 heures par mois sur 30 projets en optimisant les méthodes de travail et l'affectation des ressources.

Création d'un service d'assistance technique

Créé un service d'assistance technique ayant fourni un accompagnement informatique et résolu les incidents de plus de 10 000 clients, avec suivi des demandes dans un CRM.

5 000 h de développement cartographique pilotées

Conçu et piloté 5 000 heures de développement cartographique avec Scrum et Jira pour gérer les projets.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Projets accompagnés durant le développement · Communication et collaboration améliorées · Croissance de l'entreprise de 4 à 14 salariés · 30 projets GIS menés jusqu'à la livraison · Gain estimé : 4 000 h, équipe passée de 2 à 18

{ Conseil et direction }

Gestion de projets agile

Mener le projet à la livraison

Une personne responsable du périmètre, des dates et des conversations difficiles, pour qu'un projet ait un pilote plutôt que cinq avis.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Une norme d'historique pour 4 dépôts · Documentation soumise à son propre linter · Développement piloté par 51 initiatives écrites · Collaboration d'équipe automatisée · Gestion du temps et reporting · Analyses et rapports d'avancement réguliers · 150 h gagnées par mois sur 30 projets · Création d'un service d'assistance technique · Processus repensés : 8 000 heures gagnées · Projets accompagnés durant le développement · Communication et collaboration améliorées · Croissance de l'entreprise de 4 à 14 salariés · 30 projets GIS menés jusqu'à la livraison · Besoins métier recueillis et analysés · 5 000 h de développement cartographique pilotées · Gain estimé : 4 000 h, équipe passée de 2 à 18

{ Conseil et direction }

Stratégie produit et définition des besoins

Déterminer ce qu'il faut construire

Transformer les besoins de l'entreprise en éléments réalisables par un développeur et dire clairement ce qui ne mérite pas d'être construit.

Une identité cohérente sur tous les canaux

Développé l'identité de marque d'Engineer ApS sur les supports physiques et numériques, en reliant sa présentation à son portfolio d'ingénierie. La fondatrice fait état d’une notoriété de marque multipliée par 200 grâce à ce travail.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Erreurs réécrites sans blâmer les utilisateurs · Développement piloté par 51 initiatives écrites · Contrôle avant commit avec 22 linters · Espaces de recrutement et carrière des marins · Couche sociale : publications, fil et groupes · Portefeuille de fidélité, ligues et badges · Revendications de propriété et arbitrage · Back-office d'exploitation de la plateforme · Droits d'accès Stripe et achats intégrés · Gestion du temps et reporting · Conception UI et UX du produit de bout en bout · Système de reporting d'analyse de données · Besoins métier recueillis et analysés · 5 000 h de développement cartographique pilotées · 200 améliorations UI/UX présentées · Prévisions et investissements optimisés · Application GIS : revenus multipliés par 10 · Une communication de marque éclairée par le public

{ Conseil et direction }

Documentation technique

Écrire pour que la relève puisse suivre

Des transmissions, procédures d'exploitation et manuels qui permettent à votre équipe de faire fonctionner le système sans rappeler ses concepteurs.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Six collecteurs indiquent les entrées ignorées · Une norme d'historique pour 4 dépôts · Erreurs réécrites sans blâmer les utilisateurs · Parseur vérifiant chaque appel FFI · Contrôle des chiffres entre langues · Divergences entre hooks et contrôle qualité ·

106 contrôles sur 106 du profil PDF/UA-1 réussis · Documentation soumise à son propre linter · Développement piloté par 51 initiatives écrites · Contrôle avant commit avec 22 linters · Règle de périmètre après une fuite de secrets · 7 rôles de reporting de l'hôte en lecture seule · Programme continu d'analyse de sécurité · Moteur de 268 contrôles pour le dépôt · 578 cibles d'automatisation go-task · Migration vers Huma v2 avec parité de 100 % · 50 000 mots de documentation logicielle · 8 projets Power BI et leurs manuels · Un plan d'infrastructure pour 14 départements

{ Assistance informatique et matériel }

Assistance informatique

De l'aide pour l'informatique du quotidien

Portable, imprimante, connexion et messagerie : des problèmes réglés rapidement, en termes simples, sans mettre personne mal à l'aise.

Diagnostic et réparation de portables

Diagnostiqué et réparé le matériel de portables — écrans, charnières, claviers, pavés tactiles, cartes mères et alimentation — et résolu des incidents logiciels sous Linux, Windows et Mac.

Données récupérées : cartes SD, HDD, SSD et RAID

Récupéré des données sur de nombreux supports — cartes SD, HDD, SSD, ensembles RAID, disques externes et systèmes Mac.

4 000 réparations d'ordinateurs pilotées

Piloté 4 000 réparations d'ordinateurs pour résoudre rapidement et efficacement les problèmes matériels et logiciels.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :

Assistance 24/7 : environ 1 000 serveurs IPTV/OTT · 1 000 routeurs Wi-Fi configurés et déployés · Pilotage d'une équipe d'assistance informatique · Création d'un service d'assistance technique

{ Assistance informatique et matériel }

Réparation de matériel et récupération de données

Retrouver les fichiers d'une machine en panne

Réparer ce qui mérite de l'être et récupérer les photos, factures ou plans d'un disque qui ne fonctionne plus.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :
Diagnostic et réparation de portables · Données récupérées : cartes SD, HDD, SSD et RAID · 4 000 réparations d'ordinateurs pilotées · Pilotage d'une équipe d'assistance informatique

{ Marque et marketing }

Identité de marque, marketing et SEO

Être trouvé et donner la bonne image

Un nom, une identité visuelle et un site qui apparaissent quand quelqu'un cherche votre offre, avec un message cohérent partout où vous êtes présent.

Cette prestation intervient aussi dans des réalisations répertoriées ailleurs sur cette page :
Identité visuelle créée à partir de 2 dessins · Système de design macOS de 48 tokens · Dates identiques du sitemap corrigées · Styles allégés : de 87 KB à 31 KB · 963 blocs de métadonnées remplacés par un graphe · Système limité à 28 couleurs · 495 pages générées par un export SQLite · 27 tailles remplacées par une échelle de 6 · Développement web, SEO et contenu réunis · Une communication de marque éclairée par le public · Une identité cohérente sur tous les canaux

Comparez votre point de départ

Choisissez une réalisation comparable et identifiez ce qui diffère dans votre propre situation.

Consulter les tarifs Contactez-nous →

Discutez d'une démarche pertinente

Demandez-nous de transformer l'expérience pertinente en une démarche que vous pouvez évaluer.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/portfolio/