Engineer

Présentation complète

Engineer ApS au complet : services et exemples de réalisations, IA, plateformes, principes de travail et informations sur l'entreprise sur une page.

Découvrez ci-dessous la présentation complète : nos services et les réalisations qui les étayent, l'IA, les plateformes, nos principes de travail et les informations sur l'entreprise. Le sommaire mène aussi à chaque sujet et à notre espace de travail à Copenhague.

Explorer tous les sujets

- À propos de nous — Engineer ApS, cabinet de développement logiciel et de conseil informatique à Copenhague, créé en 2025.
- Portfolio — Parcourez l'ensemble du portfolio de réalisations d'ingénierie d'Engineer ApS, chacune accompagnée d'une analyse détaillée.
- Prestations — L'activité d'Engineer ApS — un catalogue de prestations en logiciels, données, cloud et informatique, chacune étayée par des projets réels.
- Technologies — Les technologies d'Engineer ApS : langages, plateformes de données, cloud et infrastructure, et notre façon de décider lesquelles utiliser.
- Prêt pour les agents IA — Audit de sites et d'applications : accès des robots, contenu lisible, données structurées, interfaces pour agents et mise en œuvre délimitée.
- Données d'entreprise prêtes pour les agents IA — Reliez vos données à un processus IA convenu : sources claires, contrôles qualité, autorisations et premier résultat traçable.
- Des agents IA sur mesure pour votre métier — Des agents IA pour un processus convenu, reliés à vos systèmes et évalués sur vos cas, avec contrôle humain et accès délimité.
- La sécurité de vos agents IA — Contrôles d'accès des agents IA : inventaire convenu, constats hiérarchisés, tests d'autorisations et dossier de vérification.

- Vos logiciels comme outils pour les agents IA — Exposez des opérations logicielles sélectionnées comme outils pour agents IA, avec autorisations, contrôles de contrat et déploiement planifié.
- Notre espace de travail à Copenhague

Nous bâtissons les systèmes **qui font tourner votre entreprise** — et nous en répondons.

Logiciels, données, cloud et informatique pour faire fonctionner l'ensemble — à Copenhague depuis 2025.

Engineer company

Créée en 2025

Développement logiciel et conseil en informatique

+45 50 10 61 84 · Copenhague, Danemark

Créé avec soin et amour par **D.**

Nous travaillons avec **les plateformes** sur lesquelles votre entreprise s'appuie déjà.

Apple Microsoft DigitalOcean AWS Google

Voir l'ensemble des technologies

Prêt pour les agents IA que vos clients vous envoient

Repérable

Un fichier robots qui indique ce que les agents peuvent faire, un plan du site avec des dates exactes, et chaque changement annoncé par IndexNow le jour de sa mise en ligne.

Lisible

Une copie Markdown épurée de chaque page, un fichier llms.txt servant de guide aux modèles de langage et des données structurées qui décrivent votre activité.

Utilisable

Des formulaires et des commandes accessibles, ainsi qu'une API documentée pour les opérations de l'application sélectionnées pour l'accès des agents.

Voir le fonctionnement Parlons de votre site →

Les données de votre entreprise pour étayer les réponses de vos agents IA

Rassemblées

Votre CRM, votre ERP, vos disques partagés et vos archives de messagerie réunis dans une base de données à laquelle un agent accède par un point d'entrée unique et documenté.

Nettoyées

Des identifiants convenus et des contrôles qualité relient les sources de données sélectionnées, avec une gestion des doublons et des règles de validation documentées.

Accès contrôlé

Les contrôles d'accès et la journalisation d'audit respectent les autorisations et les limites de données convenues pour l'intégration.

Voir les réalisations liées aux données Parlons de vos données →

Des agents IA conçus pour les tâches de votre métier

Connectés

Reliés à votre ERP, votre CRM, votre système de réservation et votre messagerie par des outils documentés, chacun chargé d'une tâche précise et limitée.

Évalués

Évalués sur des cas réels rédigés par vos experts, afin que chaque modification d'une instruction, d'un outil ou d'un modèle soit testée avant sa mise en service.

Supervisés

Ce que l'agent fait seul et ce qu'une personne approuve : des règles rédigées avec vous, appliquées dans les outils, avec une traçabilité étape par étape.

Voir comment les agents sont construits Parlons de votre métier →

Audit de sécurité des agents IA dans le périmètre convenu

Cartographiés

Un inventaire convenu des agents, des identifiants d'accès et des accès aux données compris dans le périmètre, conservé avec votre configuration.

Verrouillés

Chaque agent dispose de sa propre identité et du minimum de droits nécessaire ; le texte qu'il lit est traité comme des données, jamais comme des ordres.

Consignés

Chaque action est journalisée en indiquant la personne pour laquelle elle est effectuée, à un endroit que l'agent ne peut pas modifier, avec des alertes en cas d'anomalie.

Voir comment les agents sont sécurisés Parlons de vos agents →

Vos logiciels transformés en outils que les agents peuvent appeler

Décomposés

Les règles cachées dans les écrans, formulaires et procédures stockées sont identifiées, documentées et regroupées dans une seule couche.

Exposés

Chaque opération est proposée sous forme d'outil typé et documenté via une API ou un serveur MCP, agissant uniquement dans les limites des droits de son utilisateur.

Maintenus en service

Une intégration au périmètre défini peut exposer une opération à la fois, parallèlement aux écrans existants ; les modifications de l'application sont convenues après examen de ses interfaces.

Voir comment ouvrir les logiciels Parlons de vos logiciels →

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

Voir l'ensemble du catalogue

Un seul partenaire pour **38 prestations** dans six disciplines.

Plateformes et backend, données et bases de données, cloud et DevOps, conseil et direction, assistance informatique, marque et marketing.

Parcourir le catalogue

{ Ingénierie logicielle }

Décider comment construire l'ensemble

Architecture de plateformes et de solutions

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.

Voir les 20 réalisations

Notre façon de travailler

Comprendre d'abord

Nous étudions le problème avant de proposer une solution. Le cahier des charges que vous apportez est le début de cette conversation, pas sa conclusion.

Construire pour durer

La durabilité plutôt que la rapidité, la clarté plutôt que les astuces. Le système doit rester exploitable pour les personnes qui en hériteront.

Assumer la responsabilité

Toute exploitation ou assistance continue fait l'objet d'un accord dans le cadre de la mission. La transmission précise les responsabilités et la voie à suivre pour tout contact ultérieur.

{ Ingénierie logicielle }

Le moteur de votre application ou de votre site

Développement backend et API

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.

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.

Voir les 32 réalisations

{ Ingénierie logicielle }

Les écrans que les gens utilisent

Développement frontend

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.

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.

Voir les 30 réalisations

{ Ingénierie logicielle }

Un produit qui fonctionne, de bout en bout

Développement de produits full-stack

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

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.

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.

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.

Voir les 16 réalisations

{ Ingénierie logicielle }

Un site que vous pouvez mettre à jour

Développement de sites web et CMS

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.

Voir les 22 réalisations

{ Ingénierie logicielle }

Être visible quand vos clients interrogent une IA

Préparation aux agents d'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.

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.

Voir les 5 réalisations

{ Ingénierie logicielle }

Permettre à l'IA d'utiliser vos logiciels existants

Outils pour agents d'IA et serveurs MCP

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

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.

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.

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.

Voir les 5 réalisations

{ Ingénierie logicielle }

Des cartes qui répondent aux questions

Solutions GIS et géospatiales

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

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.

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.

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.

Voir les 13 réalisations

{ Ingénierie logicielle }

Rendre l'utilisation évidente

Conception UI/UX et systèmes de design

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

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.

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.

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.

Voir les 20 réalisations

{ Ingénierie logicielle }

Fonctionner dans plusieurs langues

Internationalisation et localisation

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.

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.

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.

Voir les 8 réalisations

{ Ingénierie logicielle }

Un assistant d'IA qui connaît votre métier

Développement d'agents d'IA sur mesure

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

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.

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.

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.

Voir les 5 réalisations

{ Données et bases de données }

Un endroit bien pensé pour vos données

Conception et modélisation de bases de données

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

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.

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.

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.

Voir les 14 réalisations

{ Données et bases de données }

Prendre soin de la base de données

Administration de bases de données (DBA)

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.

Voir les 9 réalisations

{ Données et bases de données }

Accélérer ce qui ralentit tout

Optimisation des performances de bases de données

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

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.

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.

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.

Voir les 6 réalisations

{ Données et bases de données }

Quitter l'ancien système

Migration et modernisation de bases de données

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

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.

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.

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.

{ Données et bases de données }

Acheminer vos données de A à B, automatiquement

Développement de pipelines de données (ETL/ELT)

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

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.

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.

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.

Voir les 14 réalisations

{ Données et bases de données }

Un endroit où tous les chiffres concordent

Conception d'entrepôts de données

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

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.

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.

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.

{ Données et bases de données }

Comprendre ce qui se passe

Analyse de données et tableaux de bord BI

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

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.

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.

Voir les 11 réalisations

{ Données et bases de données }

Pouvoir se fier aux chiffres

Gouvernance et qualité des données

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.

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.

Voir les 16 réalisations

{ Données et bases de données }

Des données que l'IA peut réellement utiliser

Préparation des données à l'IA

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

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.

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.

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.

Voir les 6 réalisations

{ Cloud, infrastructure et DevOps }

L'endroit où fonctionne votre logiciel

Infrastructure cloud et migration

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.

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.

Voir les 9 réalisations

{ Cloud, infrastructure et DevOps }

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

Infrastructure as Code

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.

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.

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.

Voir les 17 réalisations

{ Cloud, infrastructure et DevOps }

Livrer des changements sans agitation

DevOps et automatisation CI/CD

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.

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.

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.

Voir les 43 réalisations

{ Cloud, infrastructure et DevOps }

Un logiciel qui fonctionne partout de la même façon

Conteneurisation et orchestration

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.

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.

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.

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.

Voir les 5 réalisations

{ Cloud, infrastructure et DevOps }

Être informé avant vos clients

Fiabilité des services et supervision

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.

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.

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.

Voir les 23 réalisations

{ Cloud, infrastructure et DevOps }

Tout retrouver après une mauvaise journée

Sauvegarde et reprise après sinistre

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

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.

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.

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.

Voir les 5 réalisations

{ Cloud, infrastructure et DevOps }

Faire fonctionner l'essentiel

Administration système

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

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.

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.

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.

Voir les 28 réalisations

{ Cloud, infrastructure et DevOps }

Relier vos bureaux et vos équipes

Réseaux et configuration de VPN

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

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.

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.

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.

Voir les 11 réalisations

{ Cloud, infrastructure et DevOps }

Décider qui peut accéder à quoi

Sécurité et gestion des accès

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

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.

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.

Voir les 31 réalisations

{ Cloud, infrastructure et DevOps }

Limiter votre IA à ce qu'elle doit faire

Sécurité des agents d'IA

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.

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.

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.

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.

Voir les 9 réalisations

{ Conseil et direction }

Une ingénieure expérimentée à vos côtés

Direction technique et conseil

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

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.

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.

Voir les 26 réalisations

{ Conseil et direction }

Faire grandir votre équipe

Constitution d'équipes et mentorat

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.

Voir les 9 réalisations

{ Conseil et direction }

Mener le projet à la livraison

Gestion de projets agile

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

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.

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.

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.

Voir les 16 réalisations

{ Conseil et direction }

Déterminer ce qu'il faut construire

Stratégie produit et définition des besoins

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

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.

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.

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.

Voir les 19 réalisations

{ Conseil et direction }

Écrire pour que la relève puisse suivre

Documentation technique

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

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.

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.

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.

Voir les 19 réalisations

{ Assistance informatique et matériel }

De l'aide pour l'informatique du quotidien

Assistance informatique

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.

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.

Voir les 7 réalisations

{ Assistance informatique et matériel }

Retrouver les fichiers d'une machine en panne

Réparation de matériel et récupération de données

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

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.

Voir les 4 réalisations

{ Marque et marketing }

Être trouvé et donner la bonne image

Identité de marque, marketing et SEO

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.

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.

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.

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.

Voir les 11 réalisations

Dites-nous ce que vous construisez.

Pas de formulaire, pas de tunnel de conversion, pas de séquence commerciale — simplement un courriel à une personne qui le lira.

Engager la conversation

Côté serveur, nous utilisons **Linux en logiciel libre** — sans licence à payer ni rien de caché.


Ubuntu


Debian

Comment nous exploitons les serveurs

Choisissez comment le travail sera réalisé

Choisissez un partenaire d'ingénierie selon la manière dont le travail sera réalisé. Nous relions les personnes, les décisions et la réalisation nécessaires à votre projet.

Découvrir nos prestations Contactez-nous →

Posez des questions concrètes sur l'expérience

Consultez les réalisations et les méthodes de travail en complément de « Présentation complète ». Elles vous donnent des questions concrètes à poser sur l'expérience et les responsabilités.

Études de cas Contactez-nous →

Clarifiez les responsabilités

Commencez par préciser le résultat attendu et les décisions encore ouvertes. Nous pouvons déterminer où notre intervention apporterait de la valeur et ce que votre équipe devrait conserver.

Consulter les tarifs Contactez-nous →

Trouvez un mode de collaboration

Dites-nous qui a besoin du résultat et qui en aura la responsabilité ensuite. Nous pouvons discuter d'un mode de collaboration adapté à votre organisation.

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/overview/