Engineer

会社の全体像

Engineer ApSの全体像を一つのページにまとめました。各サービスと実際の仕事の例、AI、プラットフォーム、仕事の原則、会社情報を詳しく紹介します。

以下に会社の紹介をまとめています。サービスとそれを裏付ける実績、AI、プラットフォーム、仕事の原則、会社情報をご覧いただけます。目次から各テーマのページやコペンハーゲンの仕事場にも進めます。

すべてのテーマを見る

- 会社紹介 — Engineer ApSは、2025年に設立された、コペンハーゲンのソフトウェア開発・ITコンサルティング会社です。
- 実績紹介 — Engineer ApSのエンジニアリング実績を一覧でご覧いただけます。各実績には詳しい振り返りを掲載しています。
- サービス — Engineer ApSの事業内容。ソフトウェア、データ、クラウド、ITのサービスを、実際のプロジェクト実績とともにご紹介します。
- 技術構成 — Engineer ApSが使う言語、データ基盤、クラウド、インフラの技術をご紹介します。何を使うかを判断する際の考え方も説明しています。
- AIエージェントへの対応 — クローラーのアクセス、読み取れる内容、構造化データ、エージェント向けの窓口について、ウェブサイトやアプリをレビューし、範囲を定めて実装します。
- AIエージェントが使える社内データ — 合意したAI業務に社内の記録をつなぎます。データソース、品質チェック、権限を明確にし、出典をたどれる最初の回答を提供します。
- 業種に合わせた専用AIエージェント — 合意した業務に合わせ、社内システムにつながる専用AIエージェントを構築します。実例による評価、人の確認、範囲を限定したアクセスを備えます。
- 運用中のAIエージェントのセキュリティ — AIエージェントのアクセスをレビューし、制御を実装します。合意した対象一覧、優先順位付きの指摘、権限テスト、検証記録を提供します。
- ソフトウェアをAIエージェント向けツールに — 選んだソフトウェア操作を、仕様を記述したAIエージェント向けツールとして公開します。権限、API仕様の整合性検査、計画的な展開を備えます。

- コペンハーゲンの仕事場

お客様の事業を支える **システムをつくり** — その仕事に責任を持ちます。

ソフトウェア、データ、クラウド、そして全体の稼働を支えるITを、2025年からコペンハーゲンを拠点に提供しています。

---

company
Engineer
2025年設立

ソフトウェア開発・ITコンサルティングサービス

+45 50 10 61 84 · デンマーク、コペンハーゲン

心を込めて丁寧に制作： D.

お客様の事業で **すでに使われている基盤を** 私たちも扱っています。

Apple Microsoft DigitalOcean AWS Google

技術構成をすべて見る

お客様が送り込むAIエージェントに対応

見つけられる

エージェントに許可する行為を示すrobotsファイル、正確な日付を持つサイトマップ、公開当日のIndexNowによるすべての変更通知を備えます。

読める

全ページの整ったMarkdown版、言語モデル向けのllms.txtによる案内図、事業内容を示す構造化データを用意します。

使える

アクセシブルなフォームや操作部品に加え、エージェントによる利用対象として選んだアプリケーション操作に、仕様を記述したAPIを提供します。

仕組みを見る　自社サイトについて相談する →

AIエージェントが回答の根拠にできる社内データ

集約する

CRM、ERP、共有ドライブ、メールアーカイブを一つのデータベースに集め、仕様を記述した一つの窓口からエージェントが利用できるようにします。

整える

合意した識別子と品質チェックで対象のデータソースを結び付け、重複の扱いと検証ルールを文書化します。

権限を設ける

アクセス確認と監査ログは、連携について合意した権限とデータの境界に従います。

データ整備の内容を見る　自社データについて相談する →

お客様の業種の仕事に合わせたAIエージェント

接続する

それぞれの役割を限定し、仕様を記述したツールを通じて、ERP、CRM、予約システム、メールボックスに接続します。

評価する

お客様の専門家が作成した実際の事例で採点し、プロンプト、ツール、モデルの変更をすべて公開前に検証します。

監督する

自律的に行うことと、人が承認することをお客様と文書にまとめ、ツールでその制約を実施し、各手順を追跡します。

エージェントの構築方法を見る　自社の業務について相談する →

対象のAIエージェントのセキュリティレビュー

把握する

対象とするエージェント、認証情報、データへのアクセスを一覧にして合意し、設定とともに管理します。

制限する

各エージェントに固有のIDと最小限の権限を与え、読み込む文章は命令ではなくデータとして扱います。

記録する

誰のために行ったかを含め、すべての操作を、エージェントが変更できない場所に記録し、通常と異なる動作には通知を出します。

エージェントの保護方法を見る　自社のエージェントについて相談する →

お使いのソフトウェアを、エージェントが呼び出せるツールへ

解きほぐす

画面、フォーム、ストアドプロシージャに埋め込まれたルールを見つけ、文書化し、一つの層にまとめます。

公開する

各操作を、型と仕様を記述したツールとしてAPIまたはMCPサーバー経由で提供し、利用者の権限の範囲内でのみ動作させます。

稼働を続ける

範囲を定めた連携では、既存画面を維持しながら操作を一つずつ公開できます。アプリケーションの変更は、そのインターフェースを確認したうえで合意します。

ソフトウェアを外部から使える仕組みを見る　自社のソフトウェアについて相談する →

実績を読む **その背景とともに** ― 判断、実施内容、報告された成果をご覧ください。

実績一覧を見る

一社で支える **38 件のサービス** 、六つの専門分野。

プラットフォームとバックエンド、データとデータベース、クラウドとDevOps、コンサルティングとリーダーシップ、ITサポート、ブランドとマーケティング。

サービス一覧を見る

{ ソフトウェアエンジニアリング }

システム全体のつくり方を決める

プラットフォーム・ソリューション設計

まず全体像を設計し、各部分のつながりと費用を、コードを書き始める前に明らかにします。

C上にSwift 6の厳密並行処理を採用

インターフェースに必要な順序がイベントごとのタスク生成で失われると確認し、独自アクターを用いずSwift 6の完全な厳密並行処理を採用。ブロッキングCイベントループとメインアクターを、生産者1つ、消費者1つ、順序1つで接続した。

macOSアプリのイベントパイプラインに上限を設定

毎秒41 MBの増加と、36 GB端末で111 GBの圧縮ページを記録したmacOSの障害を受け、イベントバッファとログ出力に上限を設定。記録された全テスト実行では、コアのログ行が610,996から1,411へ減った。

Swift 6でネイティブmacOSクライアントを構築

固定リビジョンから静的アーカイブとしてリンクしたRustコアのCインターフェース上に、Swift 6とSwiftUIで53ファイル、11,141行のネイティブmacOSメッセージクライアントを構築した。

20件の実績をすべて見る

仕事の進め方

まず理解する

解決策を提案する前に、課題を理解します。最初にいただくご要望は、対話の結論ではなく出発点です。

仕事に責任を持つ

継続的な運用やサポートも、契約の一部として合意します。引き継ぎでは、責任の所在と、その後の連絡先を明確にします。

長く使えるものをつくる

速さより持続性、技巧より分かりやすさを大切にします。引き継いだ人が扱えるシステムであることが必要です。

{ ソフトウェアエンジニアリング }

アプリやWebサイトを裏側で動かす仕組み

バックエンド・API開発

データの保存先、処理のルール、他のシステムとの接続方法など、利用者の目に見えない部分をつくります。

未使用だったコアの半分の機能を実装

バックアップ転送、消えるメッセージ、編集・再送、確認済み招待、プロキシ、暗号化方針など、アプリケーションが未使用だったメッセージングコアの半分の機能に対応し、モックなしで実ライブラリに対して全テストを実行した。

全FFI呼び出しを検証する構文解析器を作成

手書きの仮Cヘッダーで、削除済みの3関数を呼んでもコンパイル・リンクが通り実行時に落ちた問題を受け、実際の7,308行のCヘッダーを読み、全呼び出し箇所、列挙定数、ポインター所有クラスのfinal指定を検証する構文解析器を作成した。

C上にSwift 6の厳密並行処理を採用

インターフェースに必要な順序がイベントごとのタスク生成で失われると確認し、独自アクターを用いずSwift 6の完全な厳密並行処理を採用。ブロッキングCイベントループとメインアクターを、生産者1つ、消費者1つ、順序1つで接続した。

32件の実績をすべて見る

{ ソフトウェアエンジニアリング }

実際に人が使う画面をつくる

フロントエンド開発

読み込みが速く、スマートフォンでも使え、お客様がどのブラウザーを使っていても同じように動く画面を提供します。

48トークンのmacOSデザインシステムを構築

プラットフォーム標準のガラス素材を使った48トークンのデザインシステムを構築し、マジックナンバー、固定文字サイズ、「動きを減らす」設定を無視するアニメーションを拒否する静的検査を作成した。

全17アイコンに読み上げ用の名前を付与

送信ボタンが「arrow up circle, button」と読み上げられる問題を受け、アイコンだけの全操作要素にスクリーンリーダー用の名前を付け、アイコンから8行以内のラベルを必須とする静的検査を作成した。

macOSアプリのイベントパイプラインに上限を設定

毎秒41 MBの増加と、36 GB端末で111 GBの圧縮ページを記録したmacOSの障害を受け、イベントバッファとログ出力に上限を設定。記録された全テスト実行では、コアのログ行が610,996から1,411へ減った。

30件の実績をすべて見る

{ ソフトウェアエンジニアリング }

構想から完成まで、使える製品に仕上げる

フルスタックのプロダクト開発

3社の間で受け渡す代わりに、1つのチームがアイデアのスケッチから、お客様がログインして使える製品まで担当します。

Swift 6でネイティブmacOSクライアントを構築

固定リビジョンから静的アーカイブとしてリンクしたRustコアのCインターフェース上に、Swift 6とSwiftUIで53ファイル、11,141行のネイティブmacOSメッセージクライアントを構築した。

データベース駆動の履歴書生成器を構築

19段階の冪等な処理で組み立てる23テーブルのSQLiteデータを単一の情報源として、6形式を出力する、データベース駆動の履歴書・推薦状・実績集・応募書類生成器をPythonの41モジュール、10,580行で構築した。

3言語の多言語Hugoサイトを構築

53個のpartialを含む108個のHugoテンプレートで多言語静的サイトを構築。1つのコンテンツツリーから3言語・4形式で各ページを公開し、同じツリーから7つの専門分野別サブドメインも構築した。

16件の実績をすべて見る

{ ソフトウェアエンジニアリング }

自社で更新できるWebサイト

Webサイト・CMS開発

表示が速く、検索で見つけやすく、文章を変えるたびに料金が発生せず、自社の担当者が編集できるサイトをつくります。

読みやすいTor onionミラーを公開

engineerで始まる読みやすいTor onionアドレスでサイトのミラーを公開。鍵はホスト外で生成し、このリポジトリにもノートパソコンにも置かず、4つの入口のうち計測しない唯一の入口にした。

2点の図からビジュアルアイデンティティを構築

手描きの2点の図から会社のアイコンとファビコン一式を構築し、すべての派生素材をスクリプトで再生成できるようにして、元ファイルより先に派生ファイルがコミットされた場合は検査で失敗させた。

PDF/UA-1プロファイルの106項目中106項目に合格

9文書のPDF/UA-1検証を記録し、設定済みアクセシビリティプロファイルの検査106項目中106項目に合格した一方、アーカイブ版は146規則のうち1項目で不合格だった。

22件の実績をすべて見る

{ ソフトウェアエンジニアリング }

お客様がAIに尋ねたときにも見つかるようにする

AIエージェントへの対応

お客様のAIアシスタントがサイトやアプリを見つけ、読み取り、操作できる状態を、最初の約束だけで済ませず、リリースごとに検証します。

51件の文書化した施策として開発を運営

54件の計画文書にわたる51件の施策としてサイト開発を進め、それぞれに未完了項目、測定結果、採用しなかった判断を記録。AIエージェント対応で見つかった8件中6件も、理由を記して見送った。

更新日時が共通のサイトマップを修正

エクスポートが毎回すべてのファイルを書き直すことを確認し、git履歴から更新日を取得するよう変更。176件のURL中172件が同じ更新日時だったサイトマップを修正した。

963メタデータブロックを1グラフに統合

会社の説明を1,671回繰り返していた匿名の構造化データ963ブロックを、安定したオリジン識別子から生成する16型の連結グラフ1つへ置き換えた。

5件の実績をすべて見る

{ ソフトウェアエンジニアリング }

今あるソフトウェアをAIから使えるようにする

AIエージェント用ツール・MCPサーバー

すでに動いている部分は書き直さず、既存のソフトウェア
をAIエージェントから呼び出せるツールとして公開します。

プラットフォームの運用管理画面を構築

申請、モデレーション、機能フラグ、キャッシュ、診断を扱う約40本の管理ルートと128個のコンポーネントからなる運用管理画面を構築し、データベースへ直接アクセスせずにプラットフォームを運用できるようにした。

遅い処理をRiverジョブキューへ移行

時間のかかる処理をリクエスト経路からRiverジョブキューへ移し、15個のワーカーモジュール、8件の定期タスク、20件のpg_cronジョブにより、背後の作業が続く間も応答を返せるようにした。

データベースを起点にAPI契約を生成

データベースを起点に、OpenAPI、44,076行の型付きTypeScriptクライアント、61個のモックハンドラー、UIが適用する制限を生成し、各段階に不一致で失敗する検査を設けた。

5件の実績をすべて見る

{ ソフトウェアエンジニアリング }

問いに答える地図をつくる

GIS・地理空間ソリューション

船舶、車両、区画、センサーなど、場所に関わる情
報を、検索・計測・共有できる地図にまとめます。

地図製品を最初から運用まで支える

2年間にわたり、インフラ、データ処理、地図アプリケーションを設計・開発・支援し、ソフトウェアの提供と運用保守の両方を担当した。

30件以上のGIS開発を構築からサポートまで主導

PostgreSQL、Bash、Python、JavaScript、GDAL、ArcGIS、PostGIS、Mapboxの専門知識を活用し、30件を超えるGISプロジェクトの開発・デプロイ・サポートを主導した。

GISプロジェクト30件を提供まで管理

30件のGISプロジェクトの実行を管理し、各分野で新しいソリューションを提供するための統率を担った。

13件の実績をすべて見る

{ ソフトウェアエンジニアリング }

使い方が自然にわかるようにする

UI/UX設計・デザインシステム

配置、文言、操作の流れを整え、サポートに
電話しなくても目的を果たせるようにします。

2点の図からビジュアルアイデンティティを構築

手描きの2点の図から会社のアイコンとファビコン一式を構築し、すべての派生素材をスクリプトで再生成できるようにして、元ファイルより先に派生ファイルがコミットされた場合は検査で失敗させた。

利用者を責めないエラーメッセージに改訂

実際にはプロバイダーがアプリ専用パスワードを要求していたのに、サインイン拒否を利用者の入力誤りとするエラー表示を受け、文書化した文体基準に従ってメッセージを書き直し、プロバイダー名を明示するテストを加えた。

48トークンのmacOSデザインシステムを構築

プラットフォーム標準のガラス素材を使った48トークンのデザインシステムを構築し、マジックナンバー、固定文字サイズ、「動きを減らす」設定を無視するアニメーションを拒否する静的検査を作成した。

20件の実績をすべて見る

{ ソフトウェアエンジニアリング }

複数の言語で使えるようにする

国際化・ローカライズ

日付、通貨、名前の並び順まで含め、販売先のそれぞれの市場で自然に読めるソフトウェアにします。

言語間の数値照合を構築

英語原文にある数値の欠落と、対象言語に合わない表記で失敗する言語間コンテンツ検査を構築し、指標が抜けたデンマーク語説明2件と英語式引用符を使うウクライナ語16箇所を検出した。

すべての文字列をコンテンツツリーへ移行

実際には使われていないと気づけない翻訳と、実績の3分の1だけを黙って評価する検査を発見し、利用者向け文字列をすべてPythonから3言語874ファイルのコンテンツツリーへ移した。

データベース駆動の履歴書生成器を構築

19段階の冪等な処理で組み立てる23テーブルのSQLiteデータを単一の情報源として、6形式を出力する、データベース駆動の履歴書・推薦状・実績集・応募書類生成器をPythonの41モジュール、10,580行で構築した。

8件の実績をすべて見る

{ ソフトウェアエンジニアリング }

自社の業務を理解するAIアシスタント

専用AIエージェント開発

使用中のシステムに接続し、自社の専門家が用意した事例でテストし、自社の担当者が監督できる、特定のモデルに縛られないエージェントをつくります。

6収集処理が省略した入力を明示

6つのレポート収集処理が、未読ファイル、未導入の検査、認識できなかったjournal行など、読み取れなかった入力範囲を明示するよう変更。収集不能と正常なホストが同じページになると確認し、116件の実測サンプルから閾値を定めた。

検査自体のテストを作成

正常入力しか与えない検査は、何も読まずに正常と報告するようになり得ると確認し、検査自体のテストを作成。誤字を埋めて綴り検査の検出を確かめ、テスト内の固定値ではなくデータベースからID範囲を取得した。

22静的検査のコミットゲートを構築

HTML、CSS、JavaScript、Python、YAML、Markdown、Shell、リンク、綴り、秘密情報、組版を対象に、1行で説明できる検査22個と詳しい説明が必要な検査5個からなるコミットゲートを構築し、警告扱いや行内抑制を認めない方針にした。

5件の実績をすべて見る

{ データとデータベース }

事業に合うデータの置き場をつくる

データベース設計・モデリング

実際の業務に合った構造を設計し、将来必要になる問いにも答えられるようにします。

データベース駆動の履歴書生成器を構築

19段階の冪等な処理で組み立てる23テーブルのSQLiteデータを単一の情報源として、6形式を出力する、データベース駆動の履歴書・推薦状・実績集・応募書類生成器をPythonの41モジュール、10,580行で構築した。

採用と船員キャリアの管理画面を構築

採用と船員の経歴管理の作業領域を構築。記録された実装時点では、求人、応募、資格証明、職位の昇進、乗船期間を扱う48テーブルと151のストアド関数を含む。

ロイヤルティ残高、リーグ、バッジを構築

31テーブルの台帳と67個の関数に、リーグ、バッジ、特典交換ショップを備えるロイヤルティ・評判システムを構築し、残高の行ロックで二重使用の発生余地を塞いだ。

14件の実績をすべて見る

{ データとデータベース }

データベースをきちんと保つ

データベース管理（DBA）

会社全体を支えるデータベースについて、稼働、バックアップ、気づかないうちに進む劣化まで責任を持って管理します。

1,022件のマイグレーションを差分検査で保護

新規インストールと、インストール後に全マイグレーションを適用する2通りでデータベースを構築し、不一致で失敗するCIゲートを設け、1,022件のマイグレーションにわたるスキーマの整合性を保った。

バックアップと災害復旧をコードで実装

データベースのバックアップ保持をコードで設定し、復旧可能な状態を監査して復元手順を文書化。残る不足を障害時に初めて知ることがないよう明記した。

PL/pgSQL ETL/ELTパイプライン600本を構築

複数のPostgreSQL開発・本番環境にまたがる複雑なデータ処理を効率化するため、PL/pgSQLによるETL/ELTパイプライン600本を構築した。

9件の実績をすべて見る

{ データとデータベース }

遅いところを速くする

データベース性能改善

年々重くなってきたシステムのクエリと不足するインデックスを調べ、効果のある箇所を改善します。

成長するデータモデルの時系列順キー

B-treeインデックスの断片化を抑え、クエリを高速化するため、PostgreSQL 18の時系列順UUID v7をエンティティキーとして採用した。

PostgreSQLを10倍に高速化

インデックス、パーティション、クエリを計画的に最適化してPostgreSQLを10倍に高速化し、ユーザー、テナント、地理空間、電力時系列データの処理効率を改善した。

PostgreSQL・MS SQL・BigQueryのデータウェアハウス100個を運用

主にGIS・時系列データを扱うPostgreSQL、MS SQL、Google BigQueryのデータウェアハウス100個を設計・構築・管理し、性能と拡張性を最適化した。

6件の実績をすべて見る

{ データとデータベース }

古いシステムから乗り換える

データベース移行・モダナイゼーション

業務を止めずに、老朽化したシステムから
サポートのある環境へデータを移します。

すべての文字列をコンテンツツリーへ移行

実際には使われていないと気づけない翻訳と、実績の3分の1だけを黙って評価する検査を発見し、利用者向け文字列をすべてPythonから3言語874ファイルのコンテンツツリーへ移した。

1,022件のマイグレーションを差分検査で保護

新規インストールと、インストール後に全マイグレーションを適用する2通りでデータベースを構築し、不一致で失敗するCIゲートを設け、1,022件のマイグレーションにわたるスキーマの整合性を保った。

PostgreSQLでGIS地図を10倍に高速化

データの切り替えを含め、データベースをMSSQLからPostgreSQLへ計画的に移行。処理とデータセキュリティを最適化し、地図アプリケーションの性能を10倍に向上させた。

{ データとデータベース }

データを必要な場所へ自動で届ける

データパイプライン開発（ETL/ELT）

月末に誰かが表計算ファイルをコピーしなくても、決まった時刻に必要な場所へ数値が届く仕組みをつくります。

SQLiteエクスポートから495ページを生成

読み取り専用のSQLiteエクスポートから3言語495件の実績ページを生成し、ページのアドレスをデータとして定義することで、文章を直してもURLが変わらずリンクが切れないようにした。

PostgreSQL内に独自の分析機能を構築

自動的に古いパーティションを削除するイベントテーブルと47個のPostgreSQL関数による独自の製品分析を構築し、定期変更するソルトで仮名化し、利用者の同意に基づいて動作させた。

地図製品を最初から運用まで支える

2年間にわたり、インフラ、データ処理、地図アプリケーションを設計・開発・支援し、ソフトウェアの提供と運用保守の両方を担当した。

14件の実績をすべて見る

{ データとデータベース }

部門ごとに食い違う数字をそろえる

データウェアハウス設計

営業、運用、経理の数値をまとめ、同じ会議に部門ごとに異なる答えを持ち寄る状況を解消します。

PostgreSQL・MS SQL・BigQueryのデータウェアハウス100個を運用

主にGIS・時系列データを扱うPostgreSQL、MS SQL、Google BigQueryのデータウェアハウス100個を設計・構築・管理し、性能と拡張性を最適化した。

ETL/ELTパイプライン6本を設計・運用

Google BigQuery、MSSQL、PostgreSQL、Shellスクリプト、PL/pgSQL、Transact-SQLを用いて6本のETL/ELTパイプラインを設計・実装・管理し、Python APIで効率的に処理できるようデータを統合した。

地理情報パイプラインを50倍に高速化

創業者のUtiligizeでの業務で、PostgreSQL、MS SQL、Google Cloud BigQueryのSQLとデータモデルを変更し、地理データパイプラインの50倍の改善を記録した。

{ データとデータベース }

今何が起きているかを把握する

データ分析・BIダッシュボード

意思決定を左右する少数の指標を示すダッシュボードや
レポートを、依頼しなくても更新される形で提供します。

PostgreSQL内に独自の分析機能を構築

自動的に古いパーティションを削除するイベントテーブルと47個のPostgreSQL関数による独自の製品分析を構築し、定期変更するソルトで仮名化し、利用者の同意に基づいて動作させた。

時間管理・レポートシステムを設計

大きな変更を加えずに何年も本番利用された、時間管理・レポートシステムを設計した。

分析と定期的な進捗報告を提供

分析と定期的な進捗報告をCEOへ届け、技術指標と開発状況を意思決定につなげた。

11件の実績をすべて見る

{ データとデータベース }

数字を信頼できるようにする

データガバナンス・品質管理

取り込み時に不適切なデータを検出する規則と検査
を設け、各数値の責任者と出所を明らかにします。

追跡せずに13アドレスを測定

Cookie、トラッカー、第三者サービスを使わず、Webサーバーのアクセスログのみから13アドレスを測定。書き込み時に接続元ネットワークをマスクし、21の日別集計と20の月別分析軸からなる永続アーカイブへまとめた。

公開するすべての部品のライセンスを把握

毎回のビルドで644個のRust crateのライセンス互換性を監査し、1つのcrateのライセンスを書き換えた場合と、再生成せずコアの固定リビジョンを変えた場合に検査が働くことを実証した。

言語間の数値照合を構築

英語原文にある数値の欠落と、対象言語に合わない表記で失敗する言語間コンテンツ検査を構築し、指標が抜けたデンマーク語説明2件と英語式引用符を使うウクライナ語16箇所を検出した。

16件の実績をすべて見る

{ データとデータベース }

AIが実際に利用できるデータにする

AIが使えるデータの整備

AIエージェントが記録にアクセスでき、顧客ごとの情報が一本化され、
権限を超えた情報が表示されず、回答の出所をたどれる状態にします。

963メタデータブロックを1グラフに統合

会社の説明を1,671回繰り返していた匿名の構造化データ963ブロックを、安定したオリジン識別子から生成する16型の連結グラフ1つへ置き換えた。

SQLiteエクスポートから495ページを生成

読み取り専用のSQLiteエクスポートから3言語495件の実績ページを生成し、ページのアドレスをデータとして定義することで、文章を直してもURLが変わらずリンクが切れないようにした。

1,022件のマイグレーションを差分検査で保護

新規インストールと、インストール後に全マイグレーションを適用する2通りでデータベースを構築し、不一致で失敗するCIゲートを設け、1,022件のマイグレーションにわたるスキーマの整合性を保った。

6件の実績をすべて見る

{ クラウド・インフラ・DevOps }

ソフトウェアを動かす場所を整える

クラウドインフラ・移行

実際の使用量に合わせてサーバーを構成し、月額費用を事前に把握したうえで、クラウドへの移行もクラウドからの移転も進めます。

2台目のサーバーをコードで構築

別のクラウド事業者で2台目のサーバーの構築を自動化。先にファイアウォールを作成し、起動時から保護されるようにし、両事業者のファイアウォールをREST APIから直接操作して、第三者のcollectionへの依存を避けた。

PodmanとQuadletでコンテナを運用

Dockerの既定のブリッジネットワークでのポート公開がUFWルールを迂回する場合があるため、systemd配下のPodman・Quadletでコンテナ基盤を構築。デプロイにはrootではなく、固定コマンド1つを使う非特権ユーザーを用いた。

小型サーバー上の公開サービス

利用可能メモリー464 MBを設計制約として、512 MB・1コアのホストで自社の公開サービスを運用。gitホスティング、7ドメインのWebサーバー、Tor、2つの代替プロトコルサーバー、不正接続元の遮断を備え、バックアップ自動化は別の実装として扱った。

9件の実績をすべて見る

{ クラウド・インフラ・DevOps }

バージョン管理された設定から再構築できるサーバー

コードによるインフラ管理

構成全体をファイルに記述し、特定の人の記憶だけに頼らず、まったく同じ環境を再現できるようにします。

読みやすいTor onionミラーを公開

engineerで始まる読みやすいTor onionアドレスでサイトのミラーを公開。鍵はホスト外で生成し、このリポジトリにもノートパソコンにも置かず、4つの入口のうち計測しない唯一の入口にした。

gitホスティングCIを試し2不具合を発見

gitホスティング側のCIフックを予行実行し、ファイルを読むだけでは到達できない2件の不具合を発見。解決不能な引数を入力へ渡すフォールバックと、検査先に引き継がれ19テストを失敗させるgit自身の環境変数を修正した。

すべての運用秘密情報をホストごとに分離

2ホストで1つのデッドマンズスイッチを共有すると2つのスイッチより監視能力が下がり、バックアップのパスフレーズ共有では、リポジトリが別々でも、そのパスフレーズを入手し、両リポジトリの暗号化データと鍵ファイルを読める相手が2ホストのバックアップを復号できることを確認し、ホストごとの運用秘密情報4つを分離した。

17件の実績をすべて見る

{ クラウド・インフラ・DevOps }

落ち着いて変更を届ける

DevOps・CI/CD自動化

変更を検証し、リリースし、元に戻せる仕組みを整え、金曜の深夜ではなく火曜の午後にお客様へ届けられるようにします。

4リポジトリで共通の履歴基準を維持

4つのリポジトリに対し、Conventional Commits、絵文字禁止、帰属トレーラー禁止という共通の履歴基準をコミットメッセージフックで適用し、作業方法を定める48件の指示文書とともに運用した。

公開するすべての部品のライセンスを把握

毎回のビルドで644個のRust crateのライセンス互換性を監査し、1つのcrateのライセンスを書き換えた場合と、再生成せずコアの固定リビジョンを変えた場合に検査が働くことを実証した。

未使用だったコアの半分の機能を実装

バックアップ転送、消えるメッセージ、編集・再送、確認済み招待、プロキシ、暗号化方針など、アプリケーションが未使用だったメッセージングコアの半分の機能に対応し、モックなしで実ライブラリに対して全テストを実行した。

43件の実績をすべて見る

{ クラウド・インフラ・DevOps }

どこでも同じように動くソフトウェア

コンテナ化・オーケストレーション

ノートパソコン、テストサーバー、本番環境で同じ動作になるようにまとめ、忙しい週には処理能力を増やせるようにします。

PodmanとQuadletでコンテナを運用

Dockerの既定のブリッジネットワークでのポート公開がUFWルールを迂回する場合があるため、systemd配下のPodman・Quadletでコンテナ基盤を構築。デプロイにはrootではなく、固定コマンド1つを使う非特権ユーザーを用いた。

全ユニットのsystemdサンドボックス露出を低減

この基盤が導入するすべてのユニットのsystemdサンドボックスの露出度を低減。デッドマンズスイッチのサービスを9.6 UNSAFEから1.5へ、インターネット公開のgitホスティングを8.3 EXPOSEDから1.5へ改善し、3つのユニットテンプレートで黙って無視されていた設定名の誤記を受け、収束時の構文検査を加えた。

GitHub Actions CI/CDパイプラインを構築

本番フロントエンド用のdistrolessイメージと複数環境間の昇格を備えた、GitHub ActionsのCI/CDパイプラインを構築した。

5件の実績をすべて見る

{ クラウド・インフラ・DevOps }

お客様より先に異変に気づく

サイト信頼性・監視

障害が小さいうちに知らせる監視とアラートを整え、安定した状態をどれだけ維持できているかを数値で示します。

6収集処理が省略した入力を明示

6つのレポート収集処理が、未読ファイル、未導入の検査、認識できなかったjournal行など、読み取れなかった入力範囲を明示するよう変更。収集不能と正常なホストが同じページになると確認し、116件の実測サンプルから閾値を定めた。

追跡せずに13アドレスを測定

Cookie、トラッカー、第三者サービスを使わず、Webサーバーのアクセスログのみから13アドレスを測定。書き込み時に接続元ネットワークをマスクし、21の日別集計と20の月別分析軸からなる永続アーカイブへまとめた。

macOSアプリのイベントパイプラインに上限を設定

毎秒41 MBの増加と、36 GB端末で111 GBの圧縮ページを記録したmacOSの障害を受け、イベントバッファとログ出力に上限を設定。記録された全テスト実行では、コアのログ行が610,996から1,411へ減った。

23件の実績をすべて見る

{ クラウド・インフラ・DevOps }

大きな障害のあとにすべてを取り戻す

バックアップ・災害復旧

あると信じるだけでなく、復元を試したバックアップと、
サーバーが起動しない朝に使える手順書を用意します。

すべての運用秘密情報をホストごとに分離

2ホストで1つのデッドマンズスイッチを共有すると2つのスイッチより監視能力が下がり、バックアップのパスフレーズ共有では、リポジトリが別々でも、そのパスフレーズを入手し、両リポジトリの暗号化データと鍵ファイルを読める相手が2ホストのバックアップを復号できることを確認し、ホストごとの運用秘密情報4つを分離した。

復元方法を明示したバックアップ自動化

resticのバックアップを、保持期間、整合性検査、毎月の復元テストとともに自動化し、ストレージの有効化と実環境での復元検証を別の運用手順として記録した。

バックアップと災害復旧をコードで実装

データベースのバックアップ保持をコードで設定し、復旧可能な状態を監査して復元手順を文書化。残る不足を障害時に初めて知ることがないよう明記した。

5件の実績をすべて見る

{ クラウド・インフラ・DevOps }

日々の稼働を支える

システム管理

サーバー、メール、アカウント、更新を日々管理
し、誰もが頼る仕組みを平穏に動かし続けます。

読みやすいTor onionミラーを公開

engineerで始まる読みやすいTor onionアドレスでサイトのミラーを公開。鍵はホスト外で生成し、このリポジトリにもノートパソコンにも置かず、4つの入口のうち計測しない唯一の入口にした。

6収集処理が省略した入力を明示

6つのレポート収集処理が、未読ファイル、未導入の検査、認識できなかったjournal行など、読み取れなかった入力範囲を明示するよう変更。収集不能と正常なホストが同じページになると確認し、116件の実測サンプルから閾値を定めた。

gitホスティングCIを試し2不具合を発見

gitホスティング側のCIフックを予行実行し、ファイルを読むだけでは到達できない2件の不具合を発見。解決不能な引数を入力へ渡すフォールバックと、検査先に引き継がれ19テストを失敗させるgit自身の環境変数を修正した。

28件の実績をすべて見る

{ クラウド・インフラ・DevOps }

拠点と働く人をつなぐ

ネットワーク・VPN構築

Wi-Fi、ファイアウォール、リモートアクセスを整え、どこからでも働け、権限のある人だけが接続できるようにします。

2台目のサーバーをコードで構築

別のクラウド事業者で2台目のサーバーの構築を自動化。先にファイアウォールを作成し、起動時から保護されるようにし、両事業者のファイアウォールをREST APIから直接操作して、第三者のcollectionへの依存を避けた。

20レコードのDNSゾーンをコードで照合

20レコードのDNSゾーンをCloudflare APIと宣言的に照合し、監査とBIND形式の書き出しを別の実行入口として提供。CDNプロキシは構築後、プライバシーの観点から再び無効にした。

ファイアウォールルールを位置で検証

存在するルールが実際のパケットに到達されるとは限らないため、番号付きルール一覧と稼働中のパケットフィルターチェーンを読み、ファイアウォールルールを存在だけでなく位置で検証した。

11件の実績をすべて見る

{ クラウド・インフラ・DevOps }

誰が何にアクセスできるかを決める

セキュリティ・アクセス管理

アカウント、パスワード、権限を整理し、業務に必要なアクセスだけを付与し、退職者の権限は当日に解除します。

読みやすいTor onionミラーを公開

engineerで始まる読みやすいTor onionアドレスでサイトのミラーを公開。鍵はホスト外で生成し、このリポジトリにもノートパソコンにも置かず、4つの入口のうち計測しない唯一の入口にした。

追跡せずに13アドレスを測定

Cookie、トラッカー、第三者サービスを使わず、Webサーバーのアクセスログのみから13アドレスを測定。書き込み時に接続元ネットワークをマスクし、21の日別集計と20の月別分析軸からなる永続アーカイブへまとめた。

公開するすべての部品のライセンスを把握

毎回のビルドで644個のRust crateのライセンス互換性を監査し、1つのcrateのライセンスを書き換えた場合と、再生成せずコアの固定リビジョンを変えた場合に検査が働くことを実証した。

31件の実績をすべて見る

{ クラウド・インフラ・DevOps }

AIが扱ってよいものだけを扱わせる

AIエージェントのセキュリティ

各エージェントのアクセスを必要な範囲に限定し、エージェント自身が改変できない場所に操作を記録し、安全策を信じるだけでなく検証します。

すべての運用秘密情報をホストごとに分離

2ホストで1つのデッドマンズスイッチを共有すると2つのスイッチより監視能力が下がり、バックアップのパスフレーズ共有では、リポジトリが別々でも、そのパスフレーズを入手し、両リポジトリの暗号化データと鍵ファイルを読める相手が2ホストのバックアップを復号できることを確認し、ホストごとの運用秘密情報4つを分離した。

認証情報漏出後に対象範囲の規則を記述

構成変更によって別会社の資産一覧、ファイアウォール許可、文章が混入したことを受け、リポジトリの対象範囲を定める規則を記述し、隔離した残存物も除外せず秘密情報検査の対象に残した。

秘密情報の1秒の事前検査を追加

本番実行が9番目のタスクで停止し、すでにスワップ設定が稼働ホストへ書き込まれた事態を受け、運用担当者のローカルな秘密情報に対し、収束処理と同じコードを約1秒で実行する事前検査プレイを追加した。

9件の実績をすべて見る

{ コンサルティングとリーダーシップ }

経験豊富なエンジニアが判断を支える

技術面の統率・コンサルティング

取引先、予算、設計について、知識に基づく別の見解が必要なときに、同じ側に立って判断を支えます。

4リポジトリで共通の履歴基準を維持

4つのリポジトリに対し、Conventional Commits、絵文字禁止、帰属トレーラー禁止という共通の履歴基準をコミットメッセージフックで適用し、作業方法を定める48件の指示文書とともに運用した。

テスト群を8.9秒に短縮

メインスレッドを測定し、4,017サンプル中3,989サンプルを費やす2つの呼び出しを除去。6テストで60秒の制限を超えていたテスト群を、135テストが8.9秒で通る状態に改善した。

C上にSwift 6の厳密並行処理を採用

インターフェースに必要な順序がイベントごとのタスク生成で失われると確認し、独自アクターを用いずSwift 6の完全な厳密並行処理を採用。ブロッキングCイベントループとメインアクターを、生産者1つ、消費者1つ、順序1つで接続した。

26件の実績をすべて見る

{ コンサルティングとリーダーシップ }

今いるチームを育てる

チームづくり・人材育成

採用、受け入れ、指導を通して、知識をコンサルタントとともに持ち去らせず、自社の担当者に根づかせます。

ITサポートチームを管理

1,000を超える顧客にITサポート、データ復旧、機器修理を提供するチームを管理し、サービス品質を確保した。

30プロジェクト全体で月150時間を削減

作業手順と資源配分を最適化し、30件のプロジェクト全体で月150時間を削減した。

技術サポート部門を設立

技術サポート部門を立ち上げ、10,000を超える顧客にITサポートと問題解決を提供し、CRMで問い合わせを管理した。

9件の実績をすべて見る

{ コンサルティングとリーダーシップ }

完成まで責任を持って進める

プロジェクト管理（アジャイル）

範囲、日程、難しい調整に責任を持つ担当者を置き、5つの意見が並ぶだけでなく、1人の責任者がいる状態にします。

4リポジトリで共通の履歴基準を維持

4つのリポジトリに対し、Conventional Commits、絵文字禁止、帰属トレーラー禁止という共通の履歴基準をコミットメッセージフックで適用し、作業方法を定める48件の指示文書とともに運用した。

文書を独自の静的検査で管理

文書に、引き下げのみ可能なファイル別行数上限、目次要件、実在パス検査を設けた。完了扱いの9計画中6計画に未完了項目があり、コミットを制御する14タスクがどの文書にも記載されていないことを検出した。

51件の文書化した施策として開発を運営

54件の計画文書にわたる51件の施策としてサイト開発を進め、それぞれに未完了項目、測定結果、採用しなかった判断を記録。AIエージェント対応で見つかった8件中6件も、理由を記して見送った。

16件の実績をすべて見る

{ コンサルティングとリーダーシップ }

何をつくるべきかを見極める

プロダクト戦略・要件定義

事業側の要望を開発可能な内容に変え、つくる価値がないものもはっきり伝えます。

利用者を責めないエラーメッセージに改訂

実際にはプロバイダーがアプリ専用パスワードを要求していたのに、サインイン拒否を利用者の入力誤りとするエラー表示を受け、文書化した文体基準に従ってメッセージを書き直し、プロバイダー名を明示するテストを加えた。

51件の文書化した施策として開発を運営

54件の計画文書にわたる51件の施策としてサイト開発を進め、それぞれに未完了項目、測定結果、採用しなかった判断を記録。AIエージェント対応で見つかった8件中6件も、理由を記して見送った。

22静的検査のコミットゲートを構築

HTML、CSS、JavaScript、Python、YAML、Markdown、Shell、リンク、綴り、秘密情報、組版を対象に、1行で説明できる検査22個と詳しい説明が必要な検査5個からなるコミットゲートを構築し、警告扱いや行内抑制を認めない方針にした。

19件の実績をすべて見る

{ コンサルティングとリーダーシップ }

次の担当者が使える形で書き残す

技術文書の作成

構築した人に電話しなくても自社の担当者が運用できるよう、引き継ぎ資料、運用手順書、マニュアルを整えます。

6収集処理が省略した入力を明示

6つのレポート収集処理が、未読ファイル、未導入の検査、認識できなかったjournal行など、読み取れなかった入力範囲を明示するよう変更。収集不能と正常なホストが同じページになると確認し、116件の実測サンプルから閾値を定めた。

4リポジトリで共通の履歴基準を維持

4つのリポジトリに対し、Conventional Commits、絵文字禁止、帰属トレーラー禁止という共通の履歴基準をコミットメッセージフックで適用し、作業方法を定める48件の指示文書とともに運用した。

利用者を責めないエラーメッセージに改訂

実際にはプロバイダーがアプリ専用パスワードを要求していたのに、サインイン拒否を利用者の入力誤りとするエラー表示を受け、文書化した文体基準に従ってメッセージを書き直し、プロバイダー名を明示するテストを加えた。

19件の実績をすべて見る

{ ITサポートとハードウェア }

日々のパソコンの困りごとを解決する

ITサポート・ヘルプデスク

ノートパソコン、プリンター、ログイン、メールの問題を、利用者が気後れしない平易な説明で、速やかに解決します。

ノートパソコンのハードウェアを診断・修理

画面、ヒンジ、キーボード、トラックパッド、マザーボード、電源など、ノートパソコンのハードウェアを診断・修理し、Linux、Windows、Macのソフトウェア問題も解決した。

SDカード・HDD・SSD・RAIDからデータを復旧

SDカード、HDD、SSD、RAIDアレイ、外付けドライブ、Macシステムなど、幅広い媒体からデータを復旧した。

約1,000台のIPTV/OTTサーバーを24/7で支援

中国、米国、ドイツの世界各地の顧客に向け、IPTV/OTT配信プラットフォームのインフラを24/7体制で支援し、約1,000台のサーバーと顧客所有のシステムを管理した。

7件の実績をすべて見る

{ ITサポートとハードウェア }

動かなくなった機器からファイルを救い出す

機器修理・データ復旧

修理する価値のある機器を直し、動作しなくなった
ディスクから写真、請求書、図面を取り出します。

ノートパソコンのハードウェアを診断・修理

画面、ヒンジ、キーボード、トラックパッド、マザーボード、電源など、ノートパソコンのハードウェアを診断・修理し、Linux、Windows、Macのソフトウェア問題も解決した。

SDカード・HDD・SSD・RAIDからデータを復旧

SDカード、HDD、SSD、RAIDアレイ、外付けドライブ、Macシステムなど、幅広い媒体からデータを復旧した。

コンピューター修理4,000件を管理

4,000件のコンピューター修理を管理し、ハードウェア・ソフトウェアの問題を迅速かつ適切に解決した。

4件の実績をすべて見る

{ ブランドとマーケティング }

見つけてもらい、ふさわしい印象を伝える

ブランド・マーケティング・SEO

商品やサービスを探す人に見つかり、どこで接しても同じ
印象が伝わる名前、見た目、Webサイトをつくります。

2点の図からビジュアルアイデンティティを構築

手描きの2点の図から会社のアイコンとファビコン一式を構築し、すべての派生素材をスクリプトで再生成できるようにして、元ファイルより先に派生ファイルがコミットされた場合は検査で失敗させた。

48トークンのmacOSデザインシステムを構築

プラットフォーム標準のガラス素材を使った48トークンのデザインシステムを構築し、マジックナンバー、固定文字サイズ、「動きを減らす」設定を無視するアニメーションを拒否する静的検査を作成した。

更新日時が共通のサイトマップを修正

エクスポートが毎回すべてのファイルを書き直すことを確認し、git履歴から更新日を取得するよう変更。176件のURL中172件が同じ更新日時だったサイトマップを修正した。

11件の実績をすべて見る

つくりたいものをお聞かせください。

きちんと読む担当者へのメール一通から始まり、フォームや営業への誘導、自動配信の営業メールは挟みません。

相談を始める

サーバーには **オープンソースのLinuxを採用** — ライセンス料がかからず、中身も公開されています。



Ubuntu



Debian

サーバーの運用方法

仕事の進め方を選ぶ

仕事の進め方を基に、エンジニアリングの依頼先をお選びください。プロジェクトに必要な人、判断、実施を結び付けます。

サービスを見る お問い合わせ →

経験について具体的に尋ねる

「会社の全体像」とあわせて、実績と仕事の進め方をご覧ください。経験や責任について、具体的な質問を考える材料になります。

事例 お問い合わせ →

責任を明確にする

必要な成果と、まだ決まっていないことを挙げるところから始めましょう。私たちの参加が役立つ部分と、お客様のチームが引き続き担うべき部分を整理できます。

料金を見る お問い合わせ →

一緒に仕事をする方法を探る

誰が成果を必要とし、その後に誰が責任を持つのかをお聞かせください。お客様の組織に合う進め方をご相談いただけます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/overview/