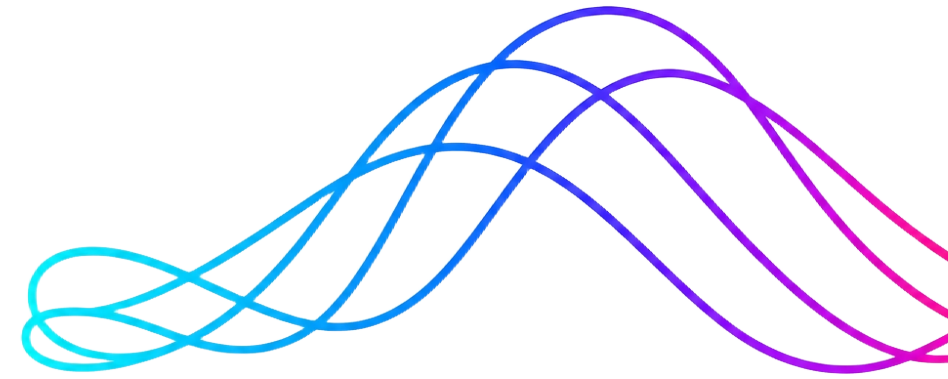


SHANNON INTRODUCTION

**Built to run.
Not just to ship.**

会社紹介資料

Products · Infrastructure · Engineering Agency



S H A N N O

会社概要

社名 シャノン合同会社 (SHANNON LIMITED LIABILITY COMPANY)

代表者 三浦 昌大 (Masahiro Miura)

設立 2026年4月 (決算月：3月)

拠点 東京都豊島区池袋2丁目17-8 / 千葉県柏市若柴178-4 / 北海道札幌市中央区北二条東3丁目2番2号マルタビル8階

資本金 9,900,000円 (取引銀行：三菱UFJ銀行 柏支店)

事業内容 プロダクト開発・運営 / インフラ構築・運用 / エンジニアリングエージェンシー

パートナー Xserverビジネスパートナー / Microsoft Cloud Partner Program

連絡先 050-1794-9651 / contact@shannon.co.jp / <https://shannon.co.jp>

3 拠点

東京・千葉・札幌

28 件

公開実績

2014年～現在

10 業種超

対応業種

金融・医療・小売・SaaS・警備など



Beyond the Launch.

リリースして終わりではなく、運用しながら価値を生み出す。その設計を、お客さまとともに進めます。



透明性 / Transparency

良い報告も悪い報告も、同じ速度で共有します。隠さない、後回しにしない。週次の進捗共有・課題管理表のオープン化・障害の即時報告として、日々の進め方に組み込んでいます。



誠実さ / Integrity

約束を守る。短期の都合で態度を変えない。曲がったことをしない。見積りは内訳付きの書面でご提示し、仕様の追加・変更は費用・納期への影響を都度示してから判断いただきます。



論理性 / Logic

感覚ではなく、データと論理で判断します。根拠のある意思決定を毎日積み重ねます。監視データ・お問い合わせ・利用統計を整理して改善のバックログを組み、優先度を定例で見直します。



長期志向 / Long-term

短期の損得より、5年・10年単位で残る設計を選びます。長く相談してもらえる会社を目指します。5年後にメンテできる構造を前提に設計し、運用手順書・設計書を継続更新します。

事業の3領域

「つくる」と「動かす」を一貫して

プロダクト運営で培った設計・運用の知見を、3つの領域で体系化して提供します。



プロダクト開発・運営

01 PRODUCT

企画から開発、運営まで。受託・レベニューシェアなど案件に合わせた形で、プロダクトづくりを一貫してご支援します。**長く運用する前提**で設計し、自社SaaS『警備サイン』『規程ログ』も自ら開発・運営しています。

- ・パーティカルSaaS開発（請負型／レベニュー型）
- ・高速PoC・プロトタイプ検証
- ・AIモダナイゼーション／AI検索最適化(AIO/GEO)
- ・自社プロダクトの運営知見を受託案件にも還元



インフラ構築・運用

02 INFRASTRUCTURE

サーバー、CI/CD、ログ、監視、セキュリティ、コスト最適化まで。基盤の**構成設計から運用まで通し**で対応し、レンタルサーバーから始まるご相談にも応じます。診断では現状維持・プランダウンの結論も率直に提示します。

- ・Web／アプリケーションサーバー設計
- ・CI/CD・ログ・監視・アラート設計
- ・クラウドコスト診断・レンタル⇄クラウド移管
- ・WordPress高速化・セキュリティ強化



エンジニアリングエージェンシー

03 AGENCY

要件整理・設計から実装、運用入りまで。SHANNONのチームが**社内の一員として期間限定で伴走**します。SESでも仕様書受託でもない、第三の関わり方で、期間伴走・継続伴走・スポットの3形態から選べます。

- ・要件整理・アーキテクチャ・技術選定
- ・実装～運用設計・内製化への引き継ぎ
- ・週次定例とSlack・GitHubでの協働
- ・スポットの設計レビュー・技術壁打ち

できること

企画・開発から基盤・運用まで、全レイヤーを提供します

3領域のサービス全体マップ。単体でも、組み合わせでもご依頼いただけます。

プロダクト開発・運営

① パーティカルSaaS開発

特定業種・業務に特化したSaaSを、企画から開発・運営まで一貫して。請負型／レベニュー型で共同開発します

② 高速PoC・プロトタイプ検証

アイデアを最短で動くプロトタイプにし、実データで仮説を検証。本開発の判断材料を短期間でそろえます

③ AIモダナイゼーション

既存システムを作り替えず、MCPでAIから安全に扱える状態へ。権限・監査を担保し、段階導入まで支援します

④ AI検索最適化（AIO/GEO）

ChatGPT等の生成AIに正しく発見・引用される状態へ。AIクローラーの受け入れ・構造化データまで整えます

インフラ構築・運用

⑤ サーバー設計・構築

Web／アプリケーション基盤の設計・構築。AWS・レンタル両対応で、案件規模に応じて構成を選定します

⑥ CI/CD・デプロイ自動化

GitHub Actions等でテスト・ビルド・デプロイを自動化し、手作業のリリースをなくしてリリース頻度と安全性を両立します

⑦ 監視・ログ基盤

死活・リソース・エラー監視とログ基盤を構築。異常は即時通知し、月額制の監視メニューもご用意しています

⑧ 診断メニュー

AWSコスト削減診断／サーバー診断・移管パック。第三者として現状維持・プランダウンの結論も提示します

エンジニアリングエージェンシー

⑨ 要件整理・設計

業務ヒアリングから機能・非機能要件の整理まで。「何を作るか」が固まる前の段階から一緒できます

⑩ アーキテクチャ・技術選定

スケール・運用・採用を見据えて技術スタックを選定し、最新技術と枯れた技術の使い分けを判断します

⑪ 実装～運用入り

バックエンド・フロントエンド・モバイルをチームで実装し、リリース後の運用設計までスコープに含めます

⑫ 内製化支援

運用手順書・設計ドキュメントを残して社内チームへ引き継ぎ。スポットの設計レビュー・技術壁打ちにも対応します

選ばれる理由



新規開発もリプレイスも

新規立ち上げから、既存システムの段階的なリプレイスまで。**引き継いだコードを解析して作り直した実績**もあり、稼働中のシステムを止めずに移行を進めます。

- ・ PHP製SaaSを18ヶ月で段階リプレイス（新機能を止めずに）
- ・ オフショア製アプリを内製品質に作り直し
- ・ ソースコード解析からのインフラ再構築



業種をまたいだ実績

金融・医療・小売・SaaS・介護・警備・教育・広告など、10業種超・28件の公開実績。**異なる業種で得た知見**を、別の案件にも活かします。詳細は後半の実績ページでご紹介します。

- ・ 地方銀行の通帳アプリ／金融商品申込サービス
- ・ 医療法人の予約基盤／介護のスマホアプリ
- ・ 商業施設のCMS／製造業のデータ基盤
- ・ 警備・大学・上場事業会社ほか



運用まで見据えた設計

インフラ構築から運用保守、内製化への引き継ぎまで同じチームで対応。**運用を前提に、はじめから設計**し、5年後にメンテできる構造を選びます。監視は「気づいて動ける」状態まで作り込みます。

- ・ CI/CD・監視・ログ基盤まで同じチームで
- ・ SLI/SLO・障害対応プロセスの設計
- ・ 運用手順書・設計書を継続更新
- ・ 内製化への移行もご相談に応じます

技術スタック

AWS Terraform Docker Kubernetes GitHub Actions Vercel Supabase Cloudflare Datadog New Relic Sentry React Next.js TypeScript
Laravel Go Node.js Python Ruby on Rails Swift Kotlin Java PostgreSQL MySQL WordPress

「人だけ出す」でも「言われたものを作る」でもない関わり方

一般的な受託開発・SESとの違い。成果に責任を持ち、運用まで切り離しません。

	SHANNON	一般的な受託開発	SES（常駐派遣）
要件確定前からの参画	◎ 要件整理から	× 仕様確定が前提	△ 指示待ちになりがち
アウトプットへの責任	◎ チームで担保	○ 仕様書の範囲のみ	× 個人スキル依存
運用設計・引き継ぎ	◎ スcopeに含む	△ 納品して終わり	× 契約外
自社プロダクトの運用経験	◎ SaaS2本を運営	△ 会社による	△ 個人による
体制の柔軟性	◎ 月単位で増減	× 契約変更が必要	○ 人月単位
仕様書の外からの提案	◎ 設計段階から提案	△ 仕様書外は出にくい	△ 判断は発注側に残る
引き継げる状態でのリリース	◎ 手順書・設計書を整備	△ 納品物の範囲による	× 属人化しがち

※ ◎＝標準で対応 ○＝一部対応 △＝限定的 ×＝非対応・別契約になりがち。一般的な契約形態との比較イメージです。実際の条件は契約内容によります。

※ 引き継げる状態でのリリース＝運用手順書・設計ドキュメントを残し、社内チームが引き取れる状態で納めること（ベンダーロックインしません）。

出典：当社サービス紹介ページ（shannon.co.jp「エンジニアリングエージェンシー」）の比較・お約束をもとに整理

SHANNONの開発プロセス

要件から運用まで通しで支える

要件を整理し、設計と実装に落とし、リリース後は監視しながら改善を継続します。

01



ヒアリング

現状・課題・制約をフラットに洗い出し、向き合うべき領域と対応範囲を合意。

02



要件定義

機能・非機能要件と技術選定を、作る前に書面で固める。見積りは内訳付きで提示。

03



設計

アーキテクチャ・データ設計。作る順番と依存関係を可視化。

04



実装・テスト

PR単位のプレビュー環境で、仕様のズレを毎週修正。テストは3段階。

05



デプロイ・監視

CI/CDで自動化し、Sentry等の監視で異常を即検知。

06



改善

運用フェーズで価値を生む。バックログ化と月次の継続改善。

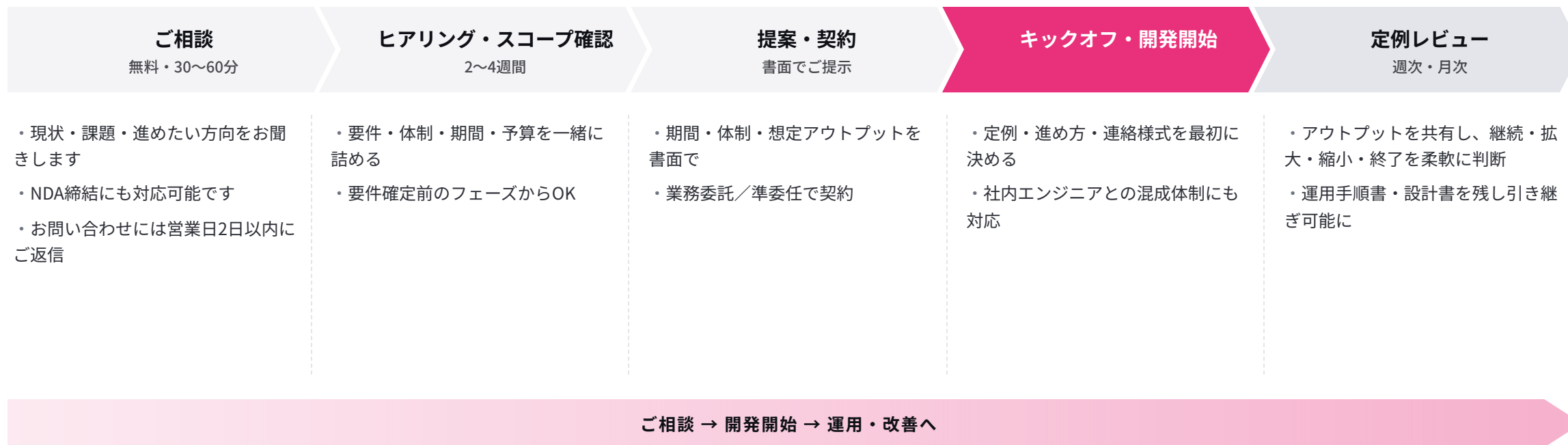
各フェーズの主なアウトプット

ヒアリング＝課題・制約の整理 / 要件定義＝要件定義書・内訳付き見積 / 設計＝設計書・実装計画 / 実装・テスト＝週次更新のプレビュー環境 / デプロイ・監視＝CI/CDと監視ダッシュボード / 改善＝優先度付きバックログ

営業日9-18時のサポート体制でインシデントに対応。監視は Sentry を中心に、Datadog / New Relic / Zabbix / CloudWatch / Grafana から案件にあわせて選定します。

ご相談から開発開始まで、最短2～4週間

最初の相談は無料。要件が固まっていない段階からご相談いただけます。



関わり方は3形態

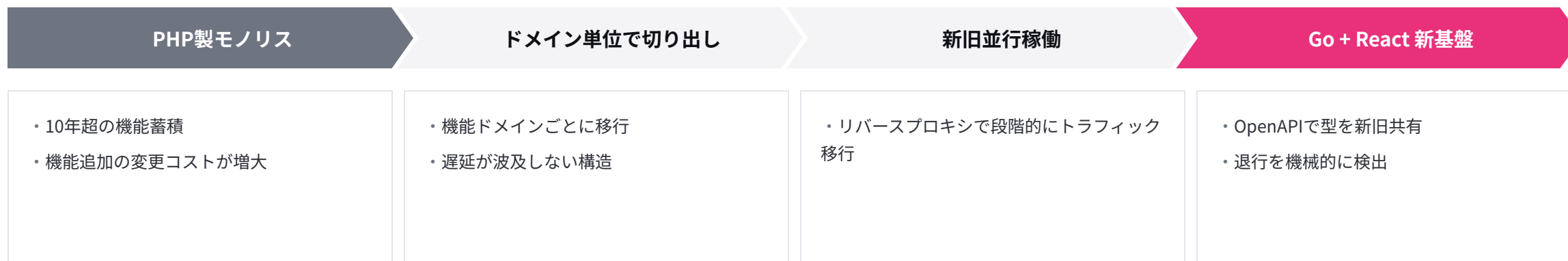
- ・ **Project（期間限定の伴走）**：3～6ヶ月単位。設計・実装・運用入りまで通しで担当
- ・ **継続伴走**：運用フェーズの定期更新・機能追加。月単位の稼働で柔軟に増減
- ・ **スポット**：設計レビュー・技術選定の壁打ち・コードベース診断。1～2週間で完結

※ 料金は要件確定後に書面でご提示します。ご相談内容は守秘義務のもと取り扱います。

HR SaaS事業者

PHP製SaaSを18ヶ月で段階リプレイス

10年超の運用で機能が積み重なったPHP製モノリスを、**新機能リリースを止めずに** Go + React + TypeScript へ移行しました。



この案件のポイント

- ・ リプレイス順序は「使用頻度 × 新基盤での再設計効果」で優先度付け
- ・ Docker + GitHub Actions で、ローカル～ステージング～本番の環境差異を最小化
- ・ 業務ロジックの整理を兼ねたユニットテストを並行整備し、動作保証ラインを引き上げ

18ヶ月

計画通りに移行進行

既存機能の大部分を新基盤へ

月次

新機能リリースを維持

リプレイスと同じスプリントで並行開発

短縮

新機能の開発リードタイム

API応答時間もPHP比で改善傾向

数百社

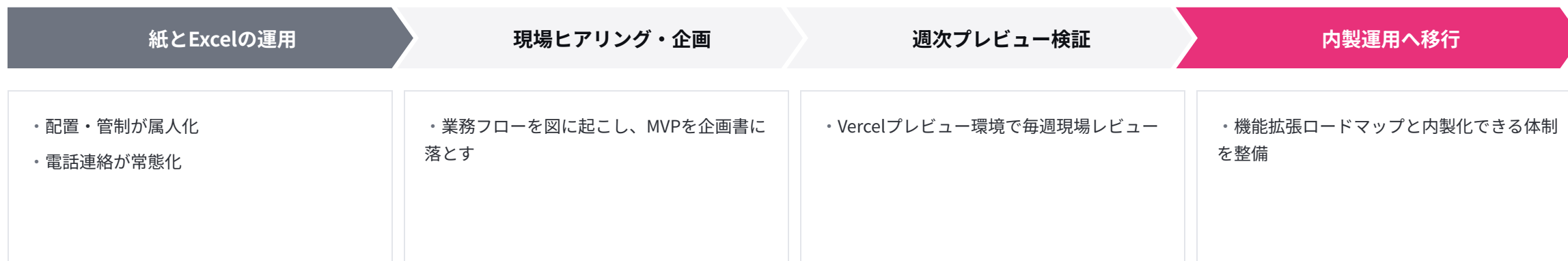
既存ユーザーへ透過的に切替

ビッグバン移行を避け、画面単位で段階切替

警備会社

管制業務・配置業務を、企画から内製化したReactアプリへ

紙とExcelの運用を、React + Supabase + Vercel で業務システム化。**現場の負担を増やさず**移行できる構造にしました。



この案件のポイント

- ・ MVPの範囲を「管制業務」と「配置業務」の2点に絞り、スコープの広がりを持ち込まない
- ・ 認証・DB・ストレージをSupabaseの最小構成でまとめ、企画段階から実物を見せて議論
- ・ 「電話で確認する」運用をアプリ完結に置き換え、配置変更の伝達コストを削減

週次

現場レビュー→翌週反映

フィードバックを翌週リリースで反映

片手

スマホ操作前提のUI

現場警備員の負担を増やさない入力設計

企画～

要件定義前から伴走

ヒアリング・業務フロー整理から支援

減少

配置調整の伝達ミス

紙・電話・Excel分散の運用を一元化

金融商品大手事業会社

申込サービスをゼロから10ヶ月で新規開発

入力項目の多い金融商品の申込を、「1画面・段階表示」で“正しく・完了させやすく”設計しました。

課題

金融商品の申込フォームは入力項目が多く、画面遷移を細かく分けると離脱が増え、まとめると入力負担が大きいというトレードオフがある。さらに監督官庁・関連機関側の要件も踏まえる必要があり、後工程で仕様が揺れると影響範囲が大きい。「正しさ」と「完了させやすさ」の両立を、要件定義の段階で骨格として固める必要があった。

成果

Laravel本体にLivewireを組み合わせ、SPA的な体験をしつつサーバ側でバリデーション・状態管理を完結。後続の機能追加でも同じフォーム設計パターンを再利用できる構造になった。

SHANNONのアプローチ

- 業務要件と監督要件を整理し、入力項目を分類・優先度付け
- 「1画面・段階表示」のフォームUIで画面遷移を増やさない方針
- 本人情報を扱う領域をVPC・IAMレベルで物理的に分離
- Tailwindで申込フォーム専用のデザインシステムを最小構成で構築
- サーバ側の状態管理で検証漏れを構造的に排除
- 監督要件と業務要件を兼ねたテストケースをCIに組み込み

10ヶ月

計画通りにリリース

要件定義から本番リリースまで予定通り進行

1画面

段階表示で申込フローを集約

サーバ側で状態管理・バリデーションを完結

想定内

離脱率が想定範囲内に収束

フォーム設計パターンは後続機能でも再利用



外食チェーン

オフショア製アプリを内製品質に作り直し

仕様書が乏しく「修正できる人がいない」状態のアプリを、Laravel + React + Inertia + TypeScript でリプレイスし、**社内エンジニアが直接触れるコードベース**に戻しました。

課題

修正のたびに別の場所が壊れ、オフショア側との時差・言語のギャップで軽微な修正にも数日～数週間。コードベースは型がなく依存関係も明文化されておらず、社内エンジニアが手を入れることも難しい。リリースのたびに障害が起き、その対応で次のリリースが遅れる悪循環が常態化していた。

成果

InertiaJSをハブにLaravel側のロジックとReact側のUIを結合し、SPA的な体験を維持したままリプレイス。社内エンジニアが直接コードに触れられる状態になり、「どこに何があるか分かる」コードベースに戻った。

SHANNONのアプローチ

- 現行アプリのユースケースを画面単位でリバースエンジニアリング
- Laravel + Inertia + React で「サーバ起点・SPA体験」の構成を採用
- TypeScript化でAPI入出力を型で固定し、退行を防止
- Dockerでローカル～本番までの再現性を担保
- コードレビュー基準とドキュメンテーション方針を策定
- リリースフローをGitHub Actionsで自動化

短縮

障害対応時間

「どこに何があるか分かる」コードベースに

短縮

機能追加リードタイム

型共有でリプレイス後の改修コストを削減

内製化

社内エンジニアが直接改修可能に

悪循環の根本原因を解消



コンプライアンス領域SaaS

内部通報システムを1週間プロト→約5人月でリリース

匿名性の設計が信頼性を左右する内部通報システムを、**小規模体制のまま高速に**立ち上げました。

課題

通報者の匿名性を完全に保護しつつ、適切な担当者にだけ内容を届ける必要がある。誰がどの通報を見られるか、どの段階で誰に通知するか、ログをどう残すか——設計レベルで詰めないとサービスの信頼性そのものが揺らぐ。さらに開発者1名体制で、まず動くものを早期に出して関係者を巻き込みたいフェーズだった。

成果

Next.js + Prisma + NextAuth でフルスタックに構築し、1週間でプロトタイプ、約5人月でリリース。匿名性とエスカレーションフローを両立する基盤として立ち上がった。

SHANNONのアプローチ

- 通報のライフサイクルとロール（通報者・受付・対応者・管理者）を整理
- Next.js (App Router) + Prisma で1週間でプロトタイプを構築し、関係者に動くものを提示
- プロト後はフィードバックを受けながら段階的にブラッシュアップ
- PMが要件整理・進行管理を担い、開発者が実装に集中できる体制を維持
- NextAuthで認証フローを実装し、ロール別アクセス制御を組み込み
- Resendで安全なメール通知（エスカレーション）、Sentryでエラー監視を整備

1週間

動くプロトタイプを提示

関係者を早期に巻き込み要件を収束

約5人月

リリースまでの総工数

PM1名＋開発者1名の少数精鋭

両立

匿名性とスピード

信頼性の設計と高速立ち上げを同時に実現

大手製造業

環境が紛失したバッチ基盤を、コード解析からゼロ再構築

既存ベンダーがインフラ環境を紛失。**稼働中のバッチを止めずに**、ソースコード解析から要件を逆算し、AWS ECS Fargate のイベント駆動構成で再立ち上げしました。

課題

顧客データ処理バッチは稼働中だったが、構成情報がほぼ手元にない状態。改修するにも、どのコードがどう繋がっているかを推測しながら進める必要があった。さらにバッチは止められないため、本番運用を維持しつつ、ソースコードからインフラ要件を逆算してゼロから再構築するという難易度の高い前提条件だった。

成果

AWS ECS Fargate + EventBridge を中核としたイベント駆動構成でゼロから再構築。「環境が消えた状態」から、本番運用を維持したまま新基盤に移行。二度と環境が失われない構成になった。

SHANNONのアプローチ

- 既存ソースコードを精読し、接続先・トリガ・データ種別などインフラ要件を逆算
- S3 → EventBridge → Fargate のチェーンで、データ到着から数秒で処理開始する構成に
- Pythonで会員情報・契約情報・使用実績・離脱判定など、データ種別ごとに処理クラスを分離
- SSM Parameter Store で接続情報・閾値を外部化
- CodeBuild で Dockerイメージ のCI/CDを整備し、再現可能な構築フローに

ゼロ

紛失環境からの再構築

ソースコード解析ベースで要件を復元

数秒

データ到着から処理開始

イベント駆動でバッチの即時性を確保

無停止

本番運用を維持したまま移行

稼働中システムを止めずに新基盤へ



業種をまたいだ実績

金融・医療・小売・SaaS・介護・警備・教育・広告など、**10業種超・28件の公開実績**。

クライアント	内容	期間
金融商品大手事業会社	金融商品申込サービスを、Laravel + Livewire でゼロから新規開発	2025.06–2026.03
地方銀行	通帳アプリを、Laravel/Lumen + React Native + Kubernetes で構築	2019.12–2020.09
医療法人	予約プラットフォームを、Laravel + AWS で要件定義から構築	2020.06–2020.12
介護事業者	介護施設のスマホアプリ+管理コンソールを一気通貫構築	2022.07–2023.05
大手小売事業者	ECサイトと管理サイトを、React + Lumen でリプレイス	2018.10–2019.07
大手製造業	顧客データバッチを、ソースコード解析からインフラ再構築まで	2024.10–2025.03
国公立大学	医科大学のWebサイト基盤を、Terraform + CloudFront で高速化対応	2024.06–2024.10
東証上場の事業会社	Webサイトリプレイスに伴うインフラ構築・運用保守	2025.01–2025.04
ヘルスケア事業者	ヘルスケアIoTデバイス連携アプリを、設計から運用設計まで12ヶ月伴走	2025.04–2026.03

ほか、HR SaaS・外食チェーン・広告・メディア・内部通報システム・AIモダライゼーションなど多数。詳細は <https://shannon.co.jp> をご覧ください。



自社プロダクト

受託の延長で生まれた、現場の道具。



警備サイン

PRODUCT 01 警備会社・建設会社向け

警備業務を一元管理するクラウドシステム。電子署名・GPS勤怠・AI配置で、紙と電話の運用から脱却し、ペーパーレス化と業務効率化を進めます。

- ・ 管制業務工数 約50%削減
- ・ 電話連絡業務ゼロ・日報自動化
- ・ 電子署名によるペーパーレス化・GPS付き勤怠管理
- ・ <https://security-sign.jp/>



規程ログ

PRODUCT 02 規程管理SaaS

社内規程・マニュアル・労使協定の作成・改定・周知・閲覧記録を一元管理するSaaS。ファイル名での版管理から卒業し、監査ログ付きの仕組みへ。

- ・ 規程テンプレート132本+AI取込
- ・ 閲覧ログ・透かし入りPDFでISMS／Pマーク監査対応
- ・ 閲覧ログ（日時・IP・本文ハッシュ）と監査用CSV書き出し
- ・ <https://kiteilog.jp/>

プロダクト『警備サイン』『規程ログ』を運用しているメンバーが、お客さまのプロダクトを担当します。設計・実装・運用の「どこで効くか」を、運用経験を踏まえて判断します。

E.O.F

