EMT
Legacy Switch2Cloud

あなたのレガシーを再創造

AIと自動化で

適応、革新、加速

ドメインを超えたグローバル企業における**メインフレーム**の問題点

脆弱性、
完全性の欠如
(古いドキュメント)

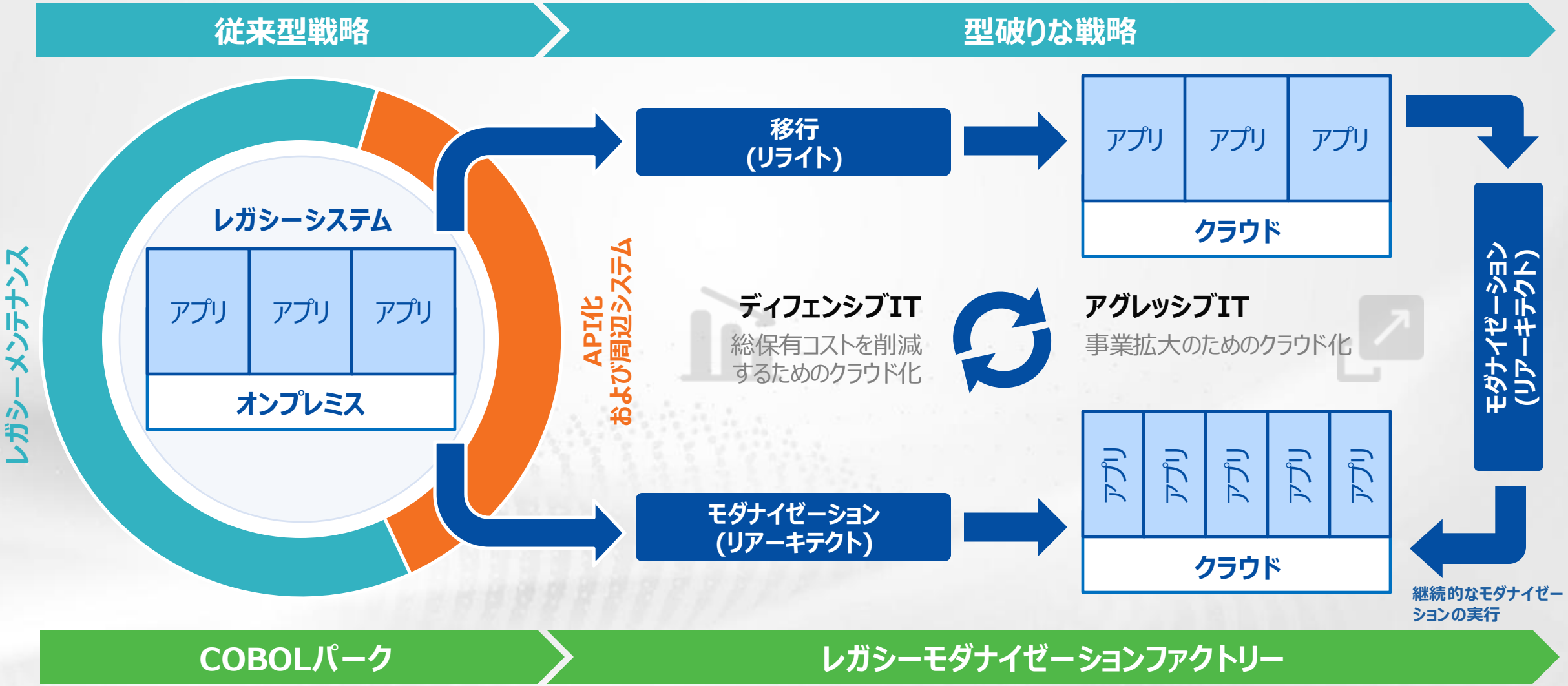
高コスト
(運用と保守)

不能
(孤立したシステム、相互
運用の欠如)

ハイリスク
(リソースストレージ)

潜在
(ビジネス変更への対応が
弱い)

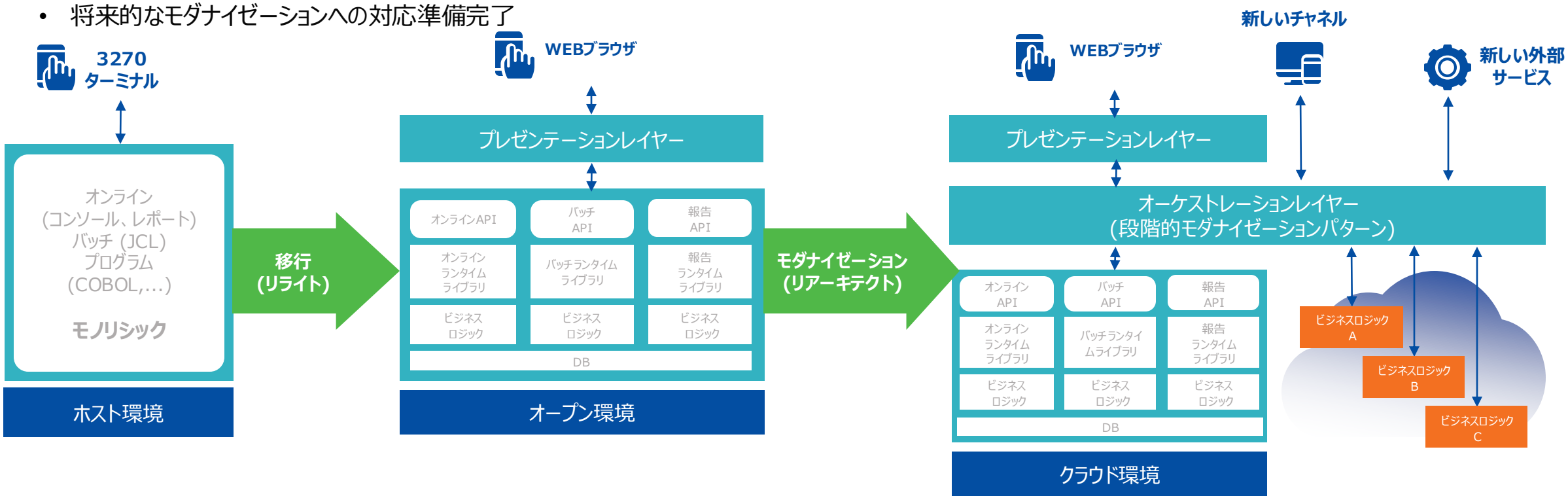
当社の変革戦略


従来型戦略
型破りな戦略
レガシーシステム
アプリ
アプリ
アプリ
オンプレミス
レガシーメンテナンス
API化
および周辺システム
移行
(リライト)
モダナイゼーション
(リアーキテクト)
ディフェンシブIT
総保有コストを削減するためのクラウド化
アグレッシブIT
事業拡大のためのクラウド化
アプリ
アプリ
アプリ
クラウド
アプリ
アプリ
アプリ
アプリ
アプリ
クラウド
モダナイゼーション
(リアーキテクト)
継続的なモダナイゼーションの実行
COBOLパーク
レガシーモダナイゼーションファクトリー

FPTでは、継続的なモダナイゼーションとの方法論を適用

FPTのマイグレーションソリューションは、移行後のアプリケーションが確実に次の条件を満たすようにします。

- 現代的なアーキテクチャフレームワークに準拠
- 保守性の向上
- 将来的なモダナイゼーションへの対応準備完了



エンジニアリングプラットフォーム

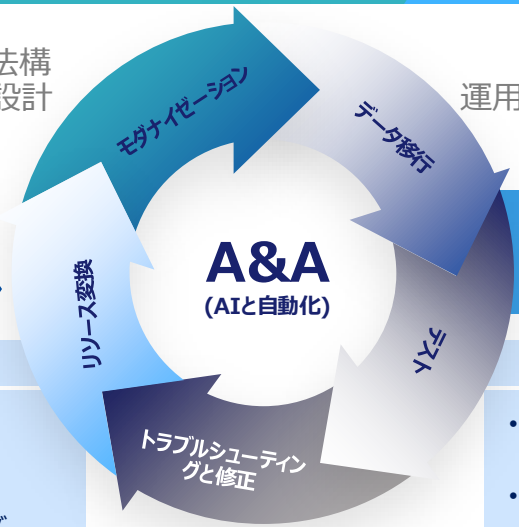
EMT
Legacy Switch2Cloud

移行ツールスイート(AIと自動化を搭載)

FPTでは、継続的なモダナイゼーションとの方法論を適用

アプリケーションのモダナイゼーションには、移行以上のものが必要であり、成功には適切なフレームワークが不可欠です。

準備フェーズ	現状分析フェーズ	開発フェーズ	
事前評価と移行計画の策定	評価、アーキテクチャ分析、および設計	カスタマイズ / ビルド / コンバージョン / 現・新比較テスト	ユーザー検証テスト・システム運用テスト
現在の資産を分析および分類し、その結果に基づいて移行計画を策定	流動資産の分析・分類、分析結果に基づく互換性のない機能の設計	ツールのカスタマイズと文法構築に基づくアーキテクチャ設計	アプリケーション品質検証 運用全体のテストを含む新システム
事前評価（ディスカバリー） 移行ポリシーと戦略	アセスメント アーキテクチャ設計 移行計画 リバースエンジニアリング	立上げ	ユーザーテスト システム運用テスト
目的	目的	目的	目的
• 移行対象システムを理解 • 移行資産を理解 • 資産分類に基づく移行ポリシーと戦略を決定	• 移行資産の特定 • As-Is/To-Beアーキテクチャを設計 • 未使用のプログラム/機能を特定 • カスタマイズを特定 • リバースエンジニアリングを実施 • 移行方法を決定とプランを作成	• チームのセットアップ • 環境の準備 • 移行支援ツールをカスタマイズ • 共通ライブラリを開発	• ユーザーテストによる品質保証 • インフラと運用を統合したシステム全体の検証



エンジニアリングプラットフォーム

EMT
Legacy Switch2Cloud

移行ツールスイート(AIと自動化を搭載)

FPTは、AIと自動化の力で
レガシーのモダナイゼーションを加速

- アセスメント
- リアーキテクチャ・リライト
- リバースエンジニアリング・リドキュメンテーション
- システムの検証と妥当性確認
- コード生成
- テスト・確認

EMT

Quick Search

Main functions

Home

Project Management

System Administration

Others

Help

Notification

Setting

AT-DEMO-2406 open

Home / Project Management / AT-DEMO-2406

Repository | Invite | New Version | Enter Project

General Information | Process Status

Overview

Name	Scope	Customer Type	Mainframe	Description
AT-DEMO-2406	Data Migration	JP	Yes	description

Migration

	Mainframe	Application Type	ONLINE	BATCH	Database
FROM	IBM/zOS	ONLINE → CICS	CICS (.BMS)	ONLINE	File (SAM/VSAM/DAM)
TO	IBM/zOS	ONLINE → CICS	Json-React	Java	Postgres

Migration and Modernization

BATCH
COBOL

Members

NuPTN
nuptn@fpt.com

マイグレーションツールスイート — ディスカバリー＆アセスメント

統計・分析を通じて、意思決定を支援するステージへ

AIと自動化の改善
- 資産分類エージェント
- コード要約・説明エージェント
- 文法提案エージェント
- 質疑応答チャットボット

資産問題分析
- デッドコード（未使用コード）
- 欠落アセット
- 重複したアセット
- 文法網羅性チェック

依存関係の抽出
- コールグラフ抽出
- 外部インターフェース抽出
- ジョブ運用
- 製品／ユーティリティの識別

アセスメントの可視化・レポート
- ダッシュボードによる可視化
- フィルタ機能付きのテーブルレポート
- Excelレポートのエクスポート

データセット分析
- データセット一覧（ファイル・データベース）
- マルチレイアウト解析
- CRUD解析

資産分類
- 資産タイプのグループ化（COBOL、JCL等）
- ドメインクラスタリング
- プロジェクトのサイジングの概要
  - ✓ コード行数（LOC）の計算
  - ✓ 複雑さの計算

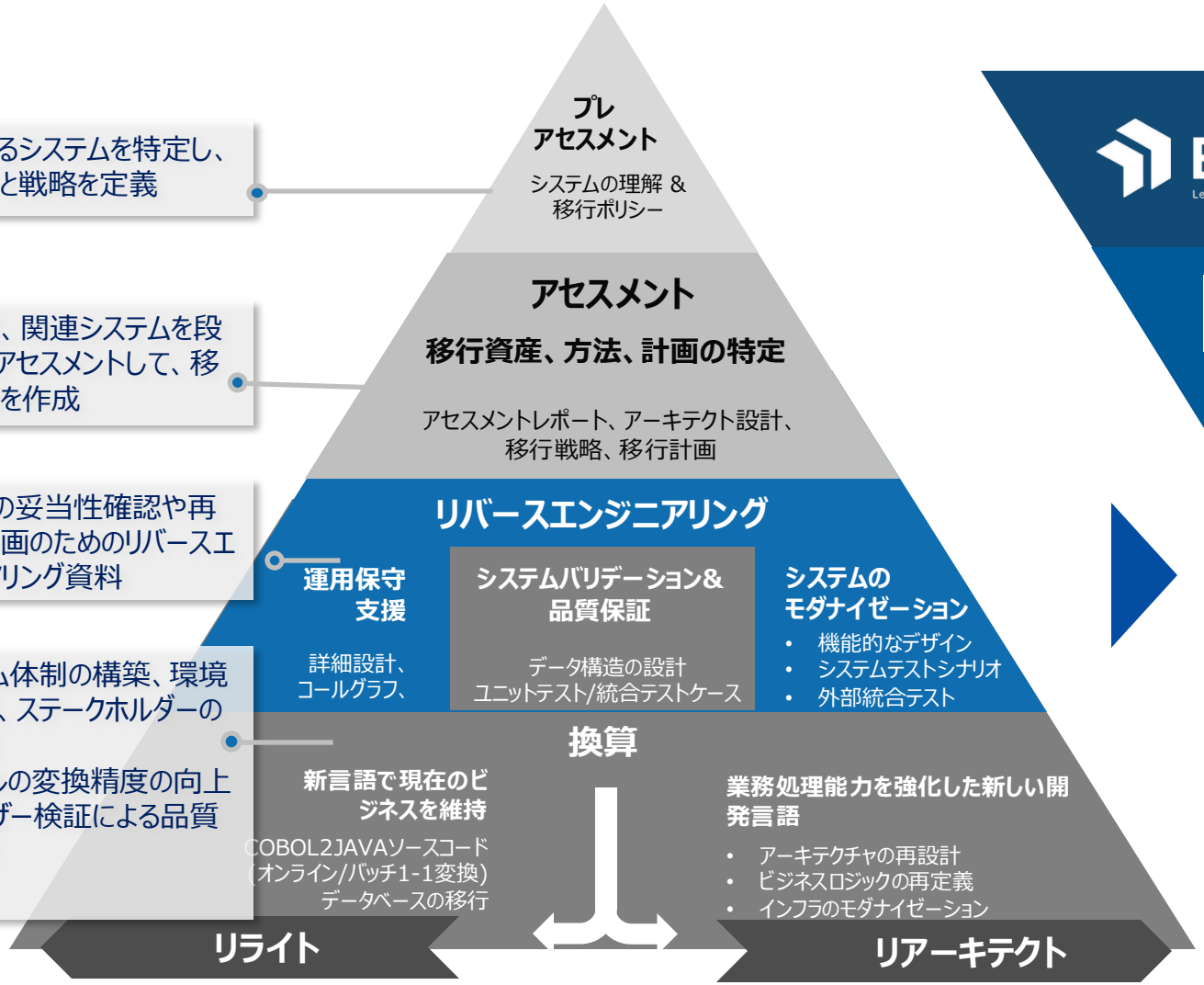
マイグレーションツールスイート — リバースエンジニアリング

移行するシステムを特定し、ポリシーと戦略を定義

全資産、関連システムを段階的にアセスメントして、移行計画を作成

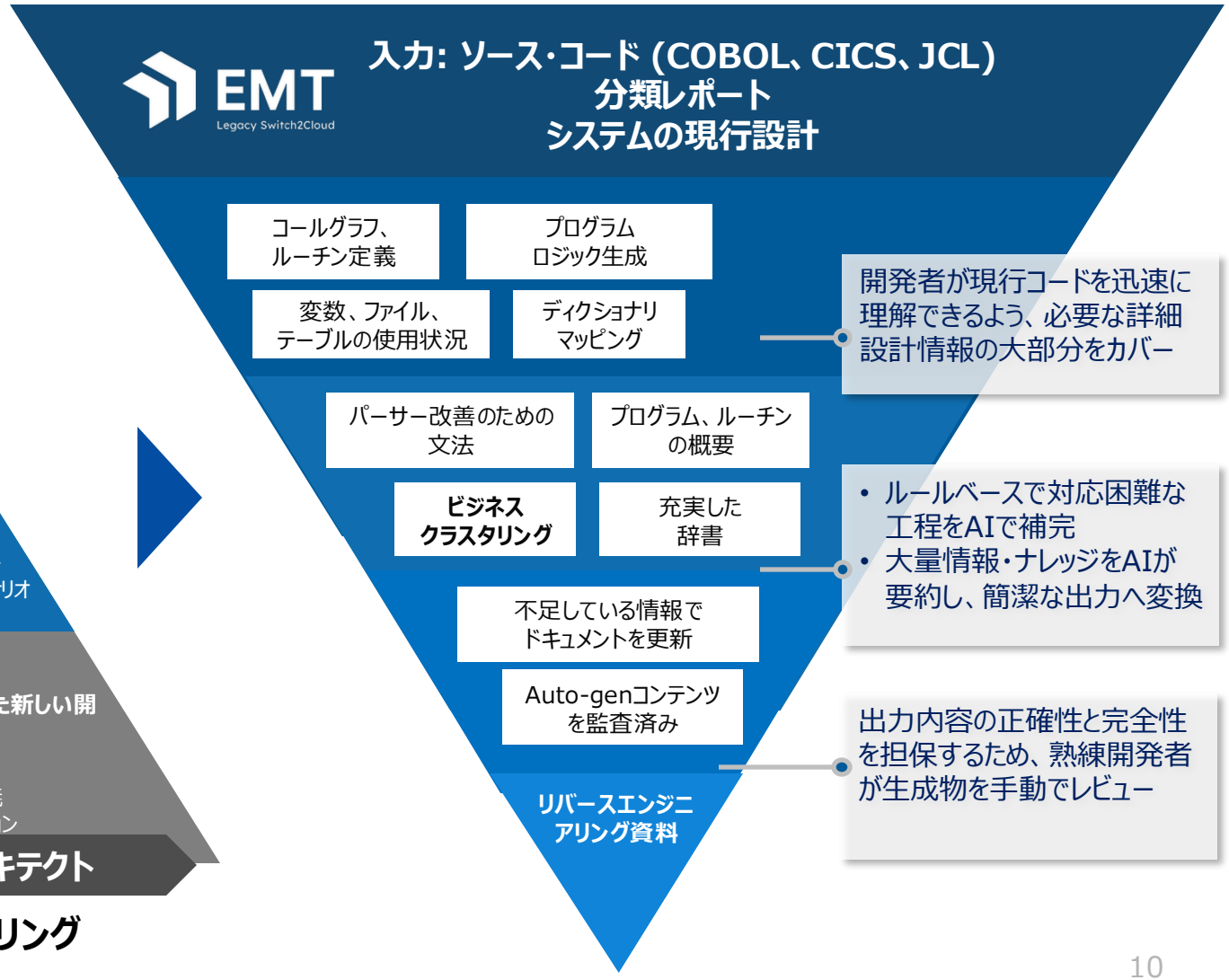
リライトの妥当性確認や再構築計画のためのリバースエンジニアリング資料

- チーム体制の構築、環境準備、ステークホルダーの整合
- ツールの変換精度の向上
- ユーザー検証による品質保証



移行とモダナイゼーションにおけるリバースエンジニアリング

リバースエンジニアリングプロセスを深く掘り下げ

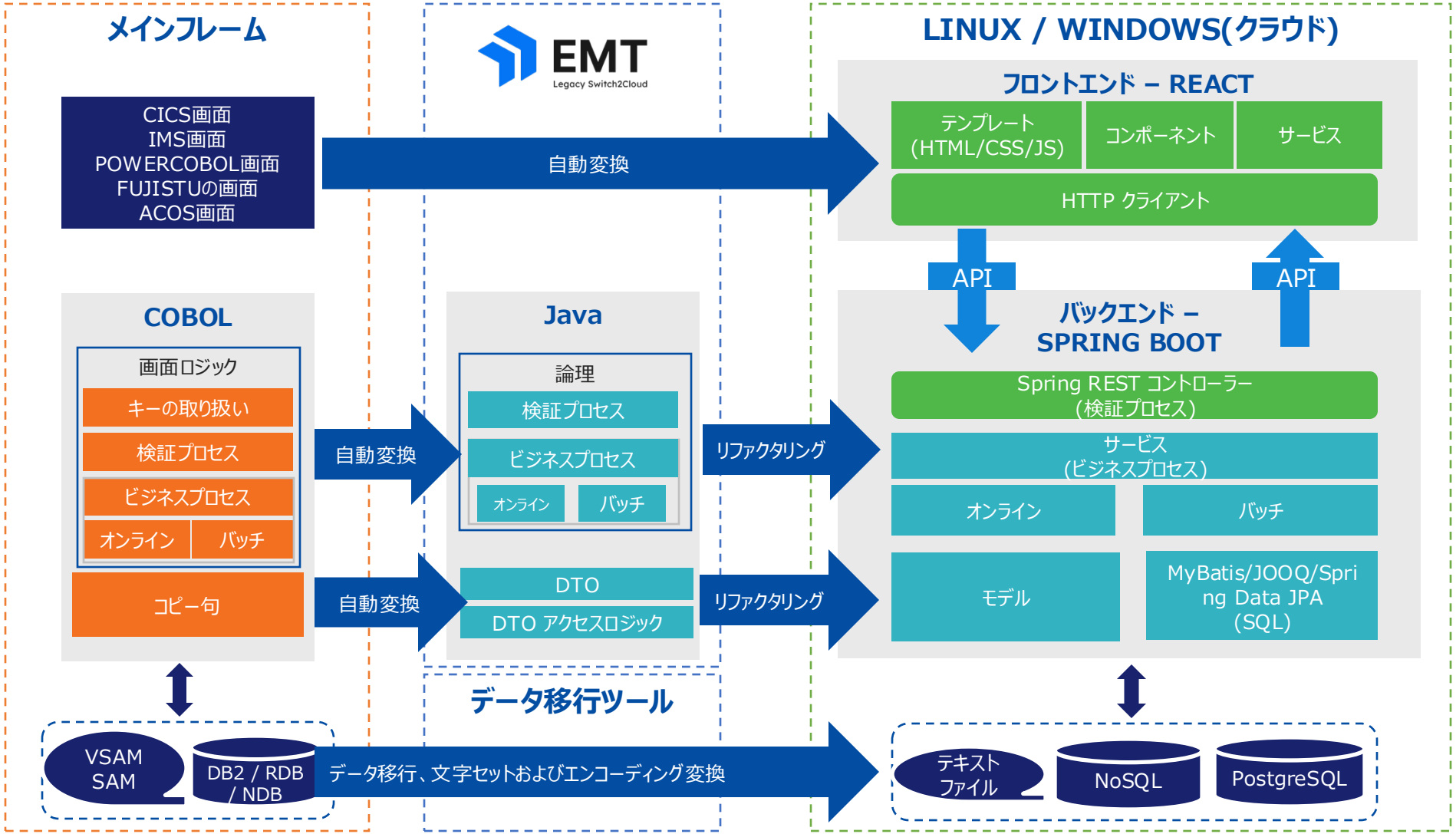


開発者が現行コードを迅速に理解できるよう、必要な詳細設計情報の大部分をカバー

- ルールベースで対応困難な工程をAIで補完
- 大量情報・ナレッジをAIが要約し、簡潔な出力へ変換

出力内容の正確性と完全性を担保するため、熟練開発者が生成物を手動でレビュー

マイグレーションツールスイート ― コード変換



リライトに基づくモダナイゼーションの方法論

1. FPTのツールを使用したコード変換(1:1)
2. ReactとFPTのMVCへのコードリファクタリング
3. Re-Make パーツ開発 (UI のモダナイゼーション)

フロントエンドの自動変換

1. HTML/CSS/JS：FPTツールによる自動変換(100%)
2. GUIイベント：ターゲットFWに応じた手動更新

移行ツールスイート – データ移行

AI 主導の移行ツールのハイライト

- AIを活用した移行前分析
- AI 駆動型の構文解析と COBOL マルチレイアウトの提案
- スマートなデータマッピングと変換

サポートされているデータベースタイプ

- SAM/VSAM
- DB2

(DBエンコーディング:EBCDIC、EBCDIK、JEF、UTF8、SJIS)

接近

- メインフレーム環境、データ構造、依存関係の分析
- 自動化ツールを使用してデータを抽出し、整合性を確保
- データ形式の変換、クレンジング、オープンシステムスキーマへのマッピング
- 移行と検証

データマルチレイアウト解析

- ルールベースとAIによるマルチレイアウト分析と認識
- インテリジェントなレイアウト抽出
- 不整合、重複、または未使用のレイアウトを強調表示します
- CRUD アクセス解析

主な利点

- コストと人的労力の削減
- 移行の迅速化とビジネスの中断の最小化
- 自動検証および調整機能により、データの一貫性、完全性、正確性を確保
- モダナイゼーションの加速

成功事例

- ~ 2500 SAM/VSAM が PostgreSQL に正常に移行されました
- ~ 100 個の DB2 テーブルが PostgreSQL に正常に移行されました

全体的なテストの概要（テストプロセスとスコープ）

アセスメントとコンバージョン

単体テスト (UT)

1. パターン比較テスト
2. 単体テスト

統合テスト(IT)

3. 内部統合テスト(ITa)
4. サブシステム間統合テスト(ITb1)
5. 外部接続テスト
6. データ移行テスト
7. インフラ試験
8. 回帰テスト

システムテスト(ST)

9. ビジネスシナリオ/サイクルテスト
10. ユーザー受け入れテスト(UAT)
11. パフォーマンス テスト
12. セキュリティテスト
13. ディザスタリカバリテスト

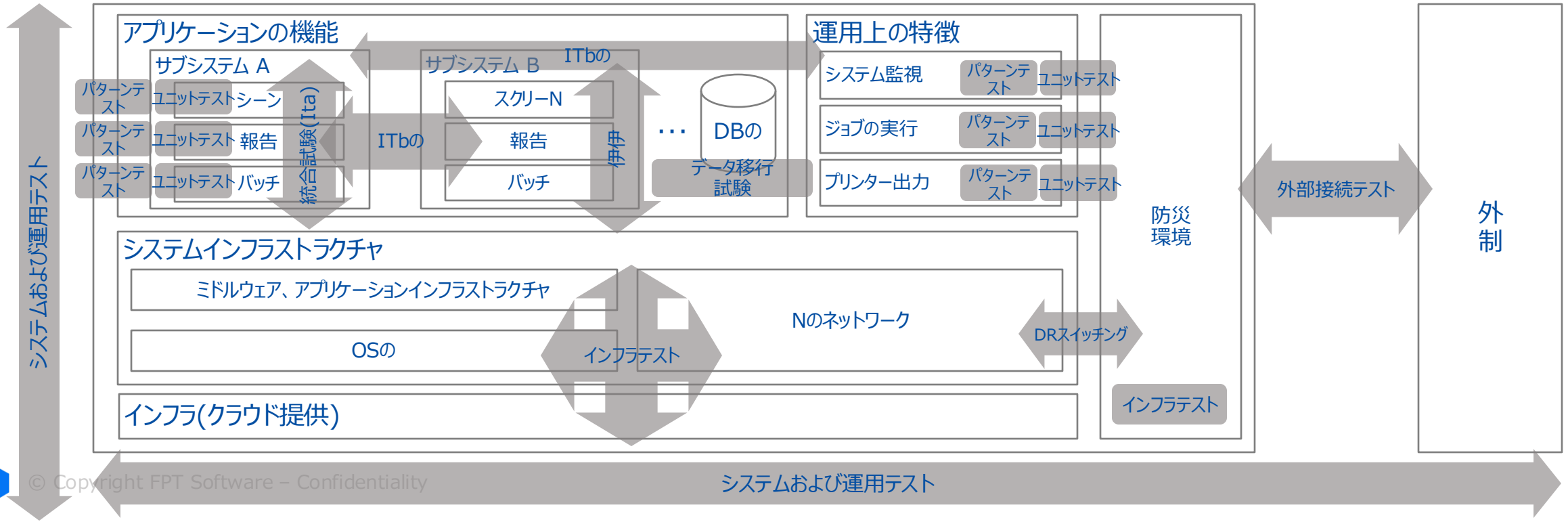
運用テスト(OT)

14. システム動作テスト
15. パラレルテスト

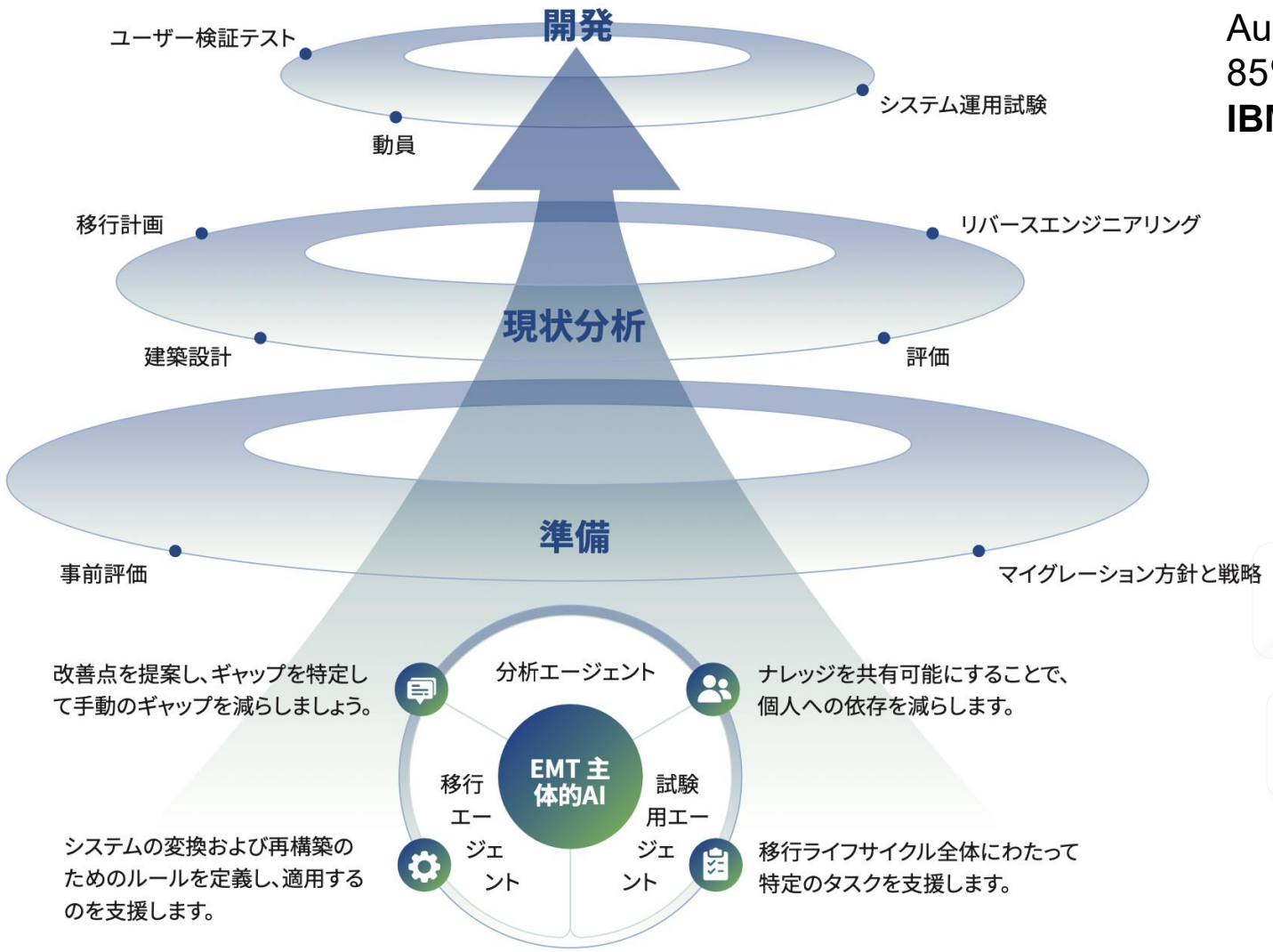
FPTの

顧客

* FPTサポート



移行ツールスイート – AI を活用したアクセラレータ



Automation & AI Agentsで75%のリバースエンジニアリングと85%の移行タスクを自動化します。
IBM メインフレームとミッドレンジを完全にサポートします。

モダナイゼーションと AI の統合:

- AI が価値を加速し、品質を向上させる
- 技術的負債を削減し、スピードを向上
- 手作業を削減し、ビジネス上のメリットを明確に
- 信頼性、顧客体験、イノベーションを強化
- モダナイゼーションは1回限りではなく進行中です
- 競争力を維持し、継続的なイノベーションを促進します

マイグレーションツールスイート ― AIを活用したアクセラレータ

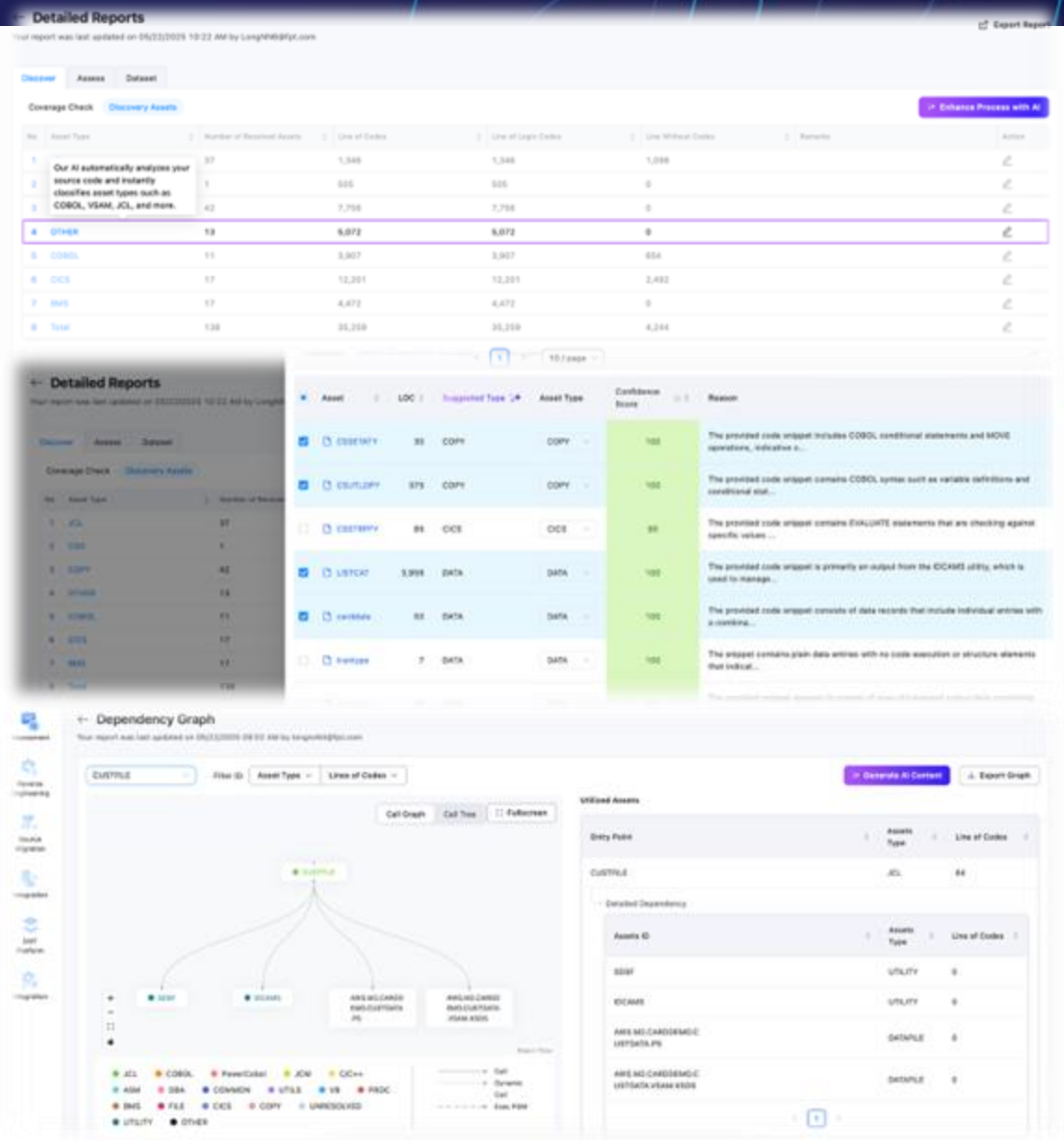
ディスカバリー&アセスメントにおけるAI

ターゲット

- **プログラム依存関係：**システム構造の可視化のためのコールグラフ／ツリー
- **欠落アセット分析：**参照されていない必要なコンポーネントを検出
- **デッドアセット分析：**未使用または廃止されたコードの特定
- **ユーティリティー一覧：**共通の再利用可能な関数を抽出
- **外部インターフェース：**API、データベース、ファイル連携のマッピング
- **重複リスト：**ロジックやコードの重複箇所を可視化
- **検討事項：**既知のリスクや障害要因のトラッキング

ベネフィット

- マイグレーションのスピードアップ
- AIによる作業精度の向上
- 対象範囲の可視化とリスクの低減
- 計画精度の向上と短期間での導入が可能



マイグレーションツールスイート ― AIを活用したアクセラレータ

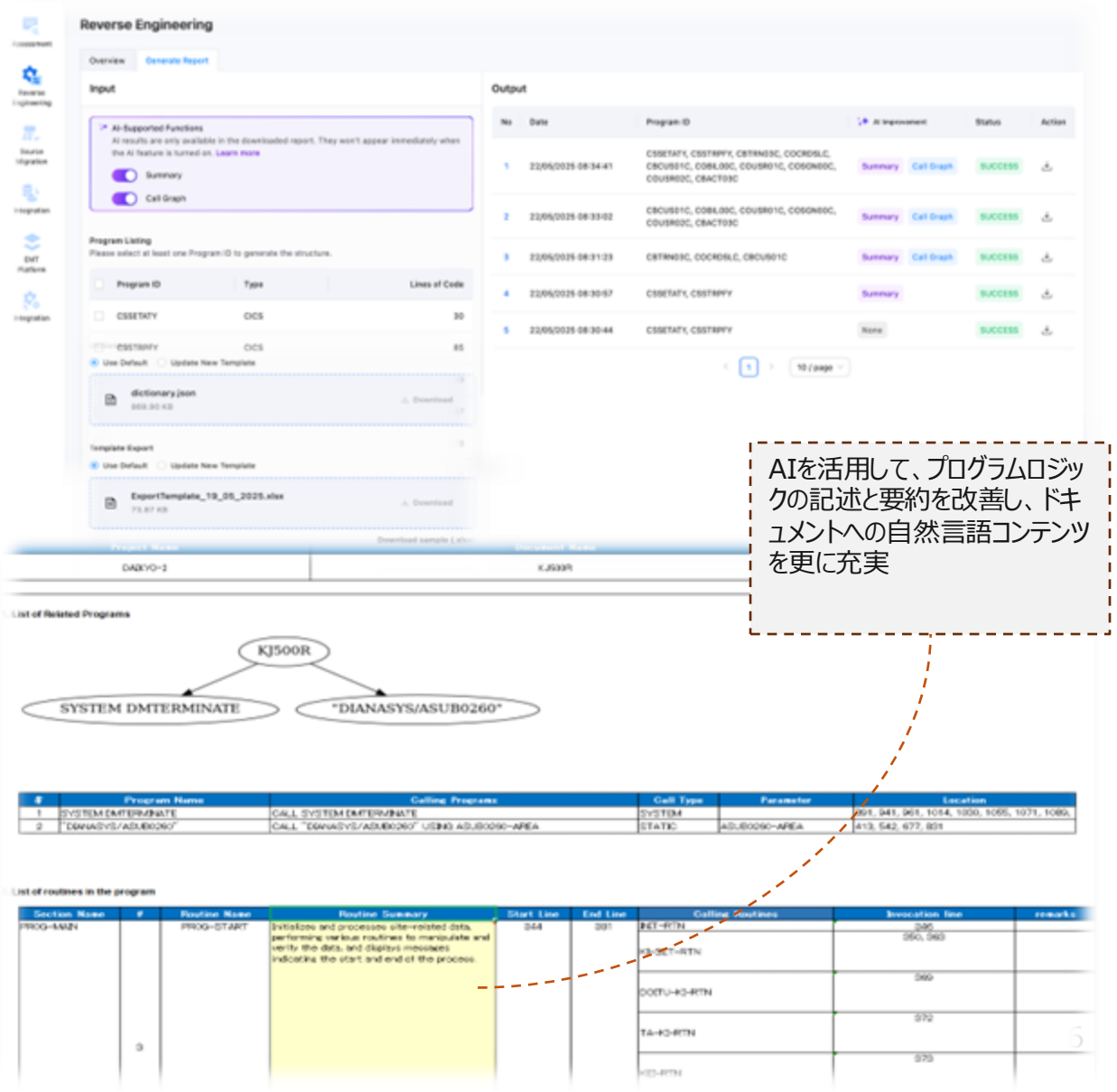
リバースエンジニアリングにおけるAI

ターゲット

- コーディング規約、ドキュメント、レガシーソースコードから辞書を構築
- 業務および技術ロジックに沿ったパースルールを定義
- 解析と翻訳
- 定義されたルールに基づきソースコードを自動解析
- コードを標準化されたドキュメント（関数一覧、処理フロー、コンポーネントマップなど）に変換

現行システム分析の必要性：リバースエンジニアリングの役割

- 設計ドキュメントの不足／モダナイゼーションの未実施
- 設計の属人化と有識者の退職
- 肥大化・複雑化したシステム構成
- システムのブラックボックス化
- 老朽化した技術基盤とブラックボックス化



マイグレーションツールスイート ― AIを活用したアクセラレータ

コード変換におけるAI

ターゲット

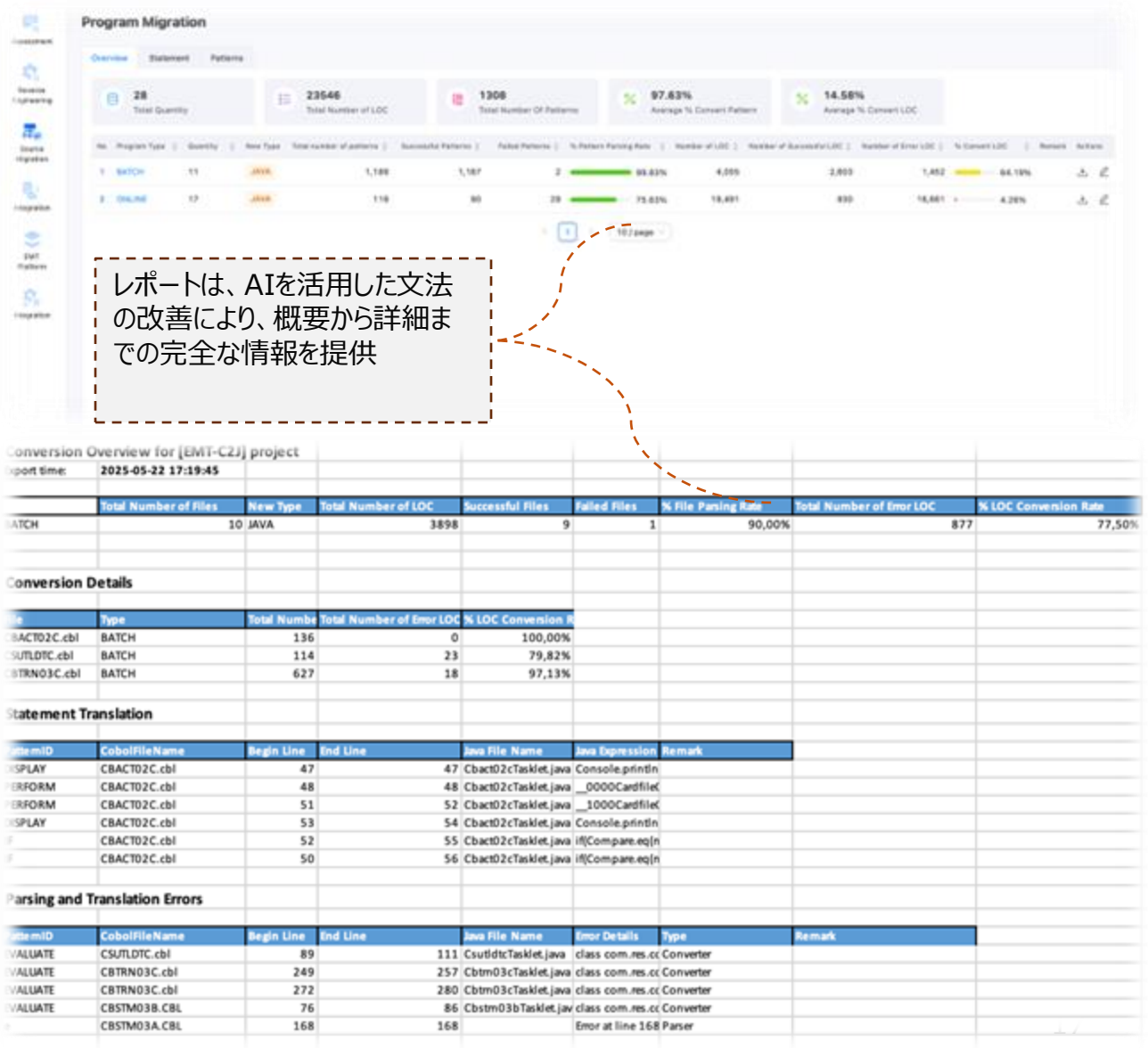
本ツールは、COBOLなどのレガシーコードをJavaなどのモダン言語へ自動移行し、FPTフレームワークに準拠した構成に変換します。

本ツールは以下の2つのスマートステップで動作します：

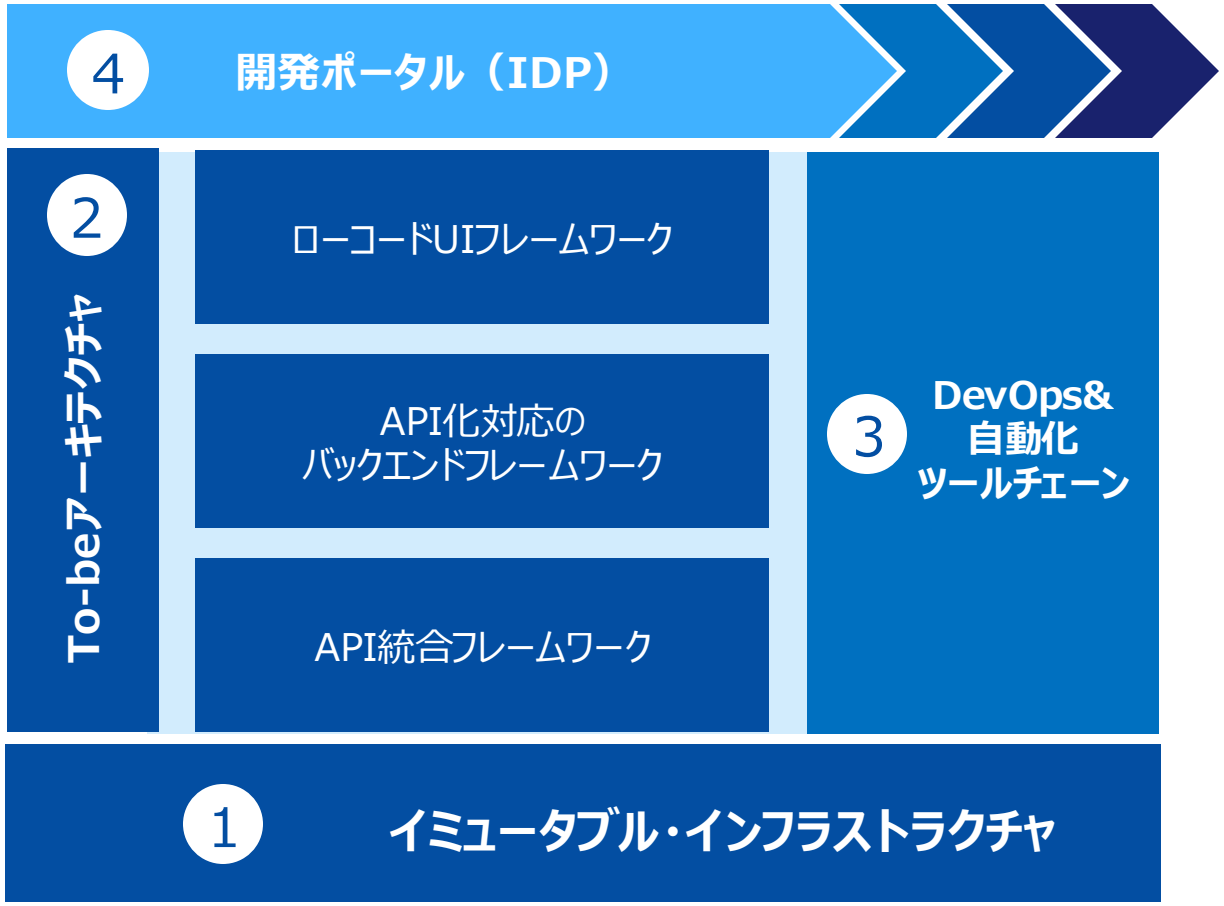
- 定義済みルールに基づき、レガシーコードを抽象的なロジックに解析
- FPTフレームワーク基準に従って、ターゲット言語および構造に変換
- 統合されたAI提案により、変換精度とカバレッジを向上させるためのパースルールを継続的に最適化

ベネフィット

- 業務ロジックを維持：重要な処理や機能を損なうことなく変換を実施
- AIによる変換精度向上：AIがルールを学習・改善し、継続的に品質を向上
- 時間とコストを大幅に削減：手動での書き換えに比べ、モダナイゼーションを迅速化
- シームレスなエンドツーエンドのマイグレーション：アセスメント → リバースエンジニアリング → 変換 → テストまで、一貫した移行プロセスを提供



エンジニアリングプラットフォーム



1 イミュータブル・インフラストラクチャ

IaC（Infrastructure as Code）に対応したクラウドベースの環境であり、運用面では「修正するのではなく、終了して新しいものに置き換える」アプローチを採用します。問題が発生した場合は、該当インスタンスを修正するのではなく、新しいインスタンスに置き換えます。

2 To-Beアーキテクチャ

エンジニアがビジネス機能を開発する際に活用できる標準的なフレームワークです。ソースコードのテンプレートや組み込みコンポーネントを含んでいます。

3 DevOpsおよび自動化ツールチェーン

品質管理、自動デプロイ、モニタリング、セキュリティおよびコンプライアンスを支援するためのソリューション一覧

4 開発ポータル

エンジニアや運用担当者が実装および運用プロセスと簡単に連携できるようにするセルフサービス型ポータル

エンジニアリングプラットフォーム

4 開発ポータル（IDP） Backstage

2

ローコード
UIフレームワーク

API化対応の
バックエンド
フレームワーク

API統合
フレームワーク

3

バージョン管理

github

Gitlab

ビッツバケット

開発者プレーン

CIパイプライン

Githubのアクション

Jenkins

Gitlab CI

Azure DevOps

AWS コードパイプライン

CDプレーン

argo

インフラCI/CD

spacelift

CI/CDプレーン

オブザーバビリティ

データ・データ・エンド

dynatrace

elk

OpenTelemetry

new relic

JAEGER

モニタリング プレーン

セキュリティ

Vault

sonarqube

セキュリティプレーン

プラットフォーム オーケストレーター

クベベラ

クロスプレーン

IAPの
ワークフロー

IaCツール

Terraform

AWS CDK

Pulumi

AI

Hugging Face

OpenAI

n8n

プラットフォームプレーン

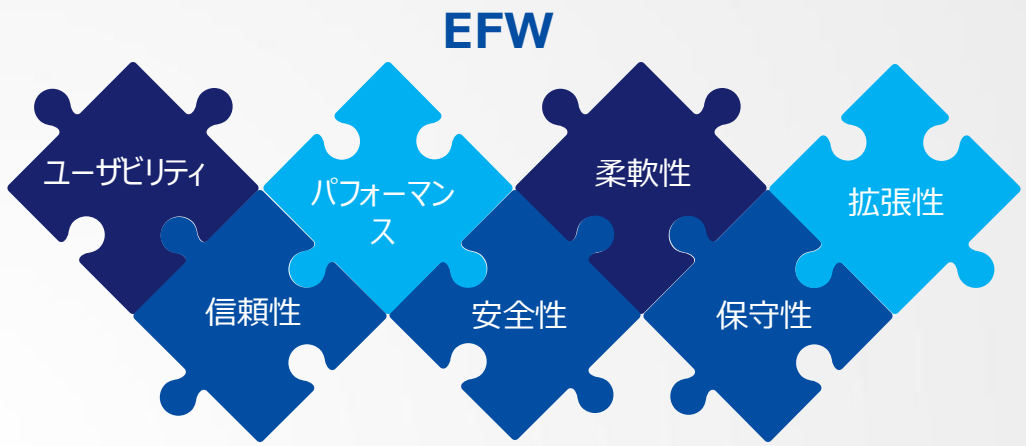
1 クラウドネイティブ／イミュータブルインフラ aws Azure Google Cloud Platform kubernetes

To-be アーキテクチャ

最新のWebアプリケーションは4層アーキテクチャを採用。

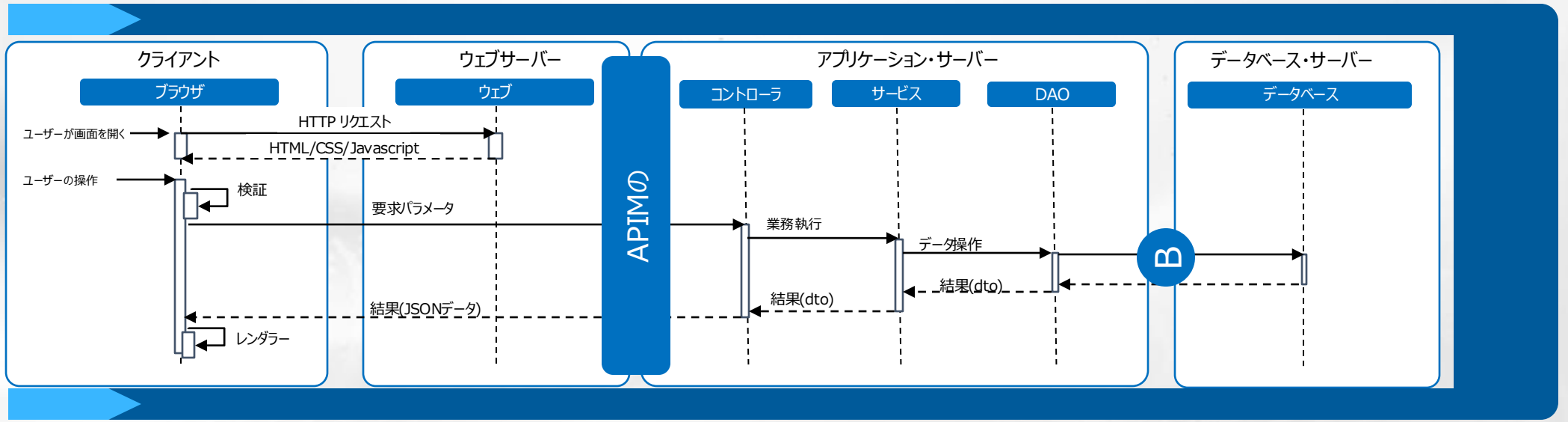
EFWもこのパターンに従います：

- クライアント層（ブラウザー クライアント）
- Web 層（Web UI）
- ステートレスアプリケーション層
- データベース層



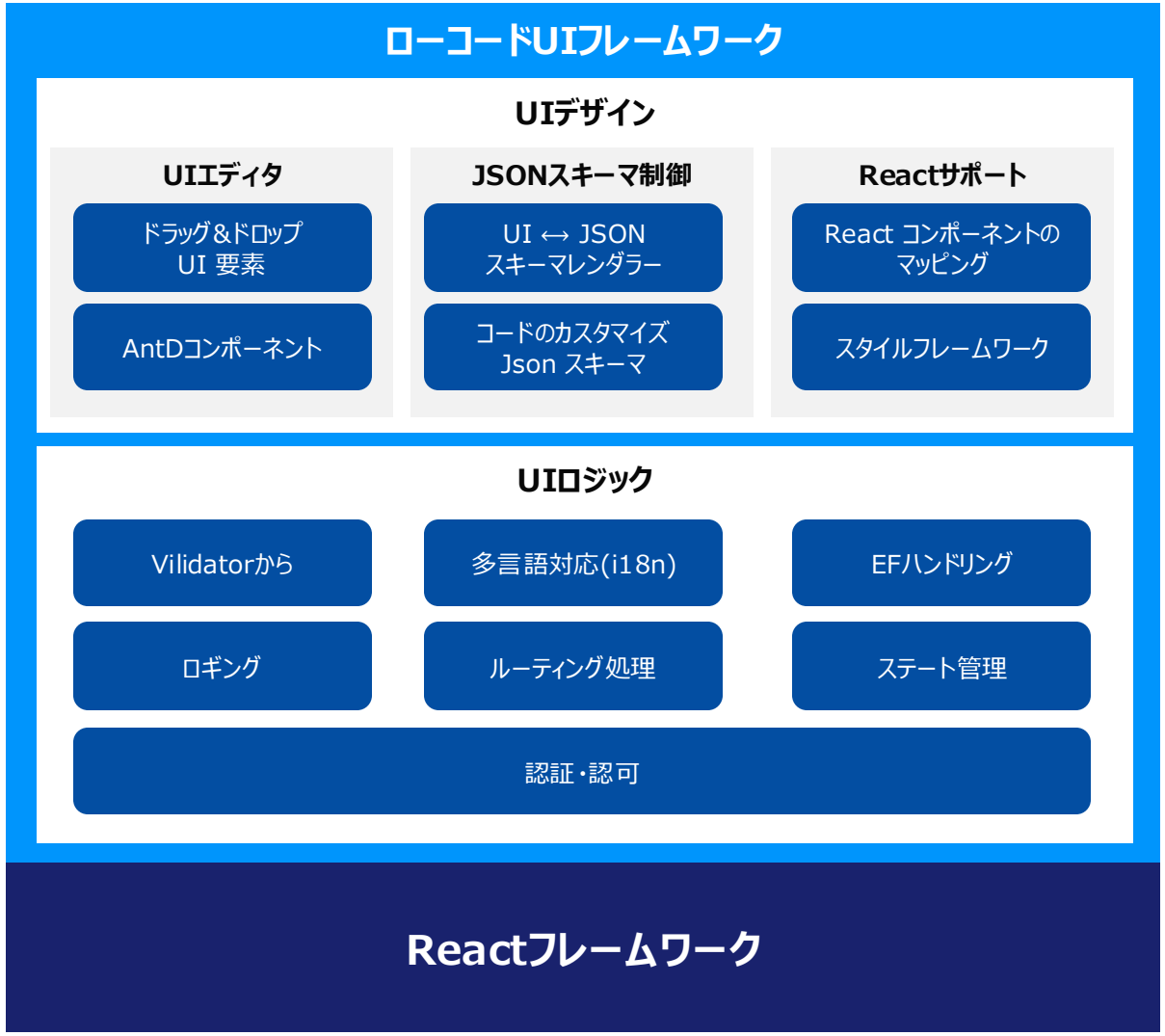
A ローコードUIフレームワーク

B API化対応のバックエンドフレームワーク

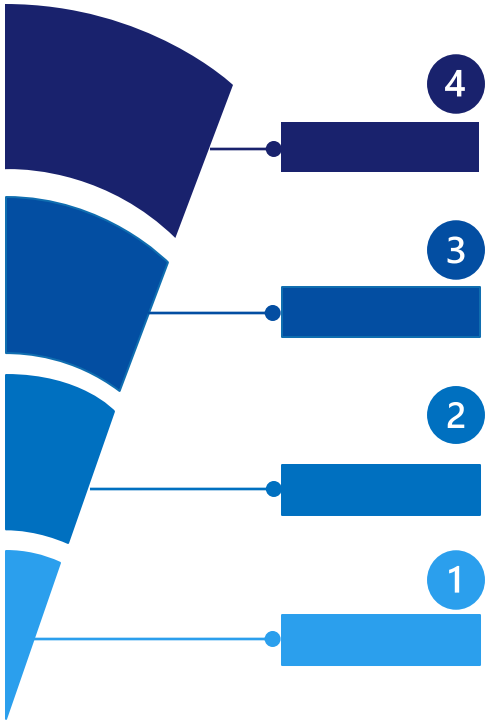


C API統合プラットフォーム

ローコードUIフレームワーク



ローコードUIフレームワークの利点



提供価値

- ✓ デリバリーのスピードアップ
- ✓ 開発工数の削減
- ✓ スケーリングとメンテナンスの容易化
- ✓ 移行リスクの低減
- ✓ 高い柔軟性と再利用性

移行時間

- 安定したJSON UI構造
- ReactとVueの移行をサポート

実装時間

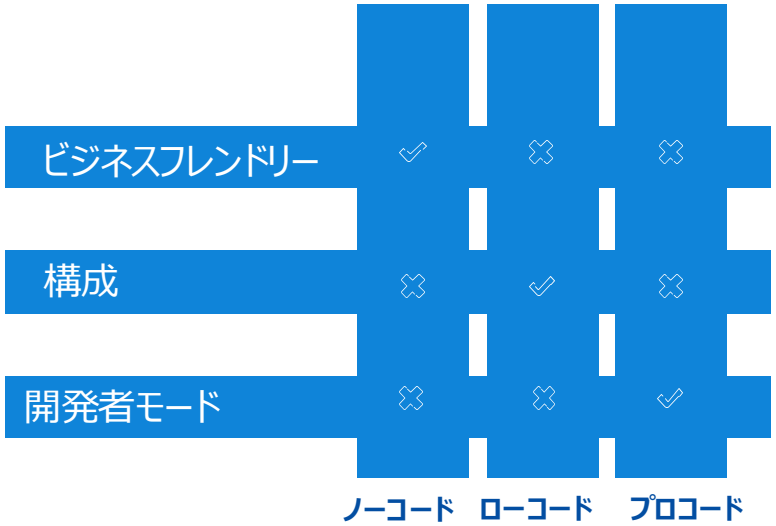
- 組み込みのセキュリティ、検証
- UIとロジックの分離

デザインタイム

- ドラッグ&ドロップの高速UIデザイン
- 再利用のためのJSONベースの構成

ローコードUIフレームワーク

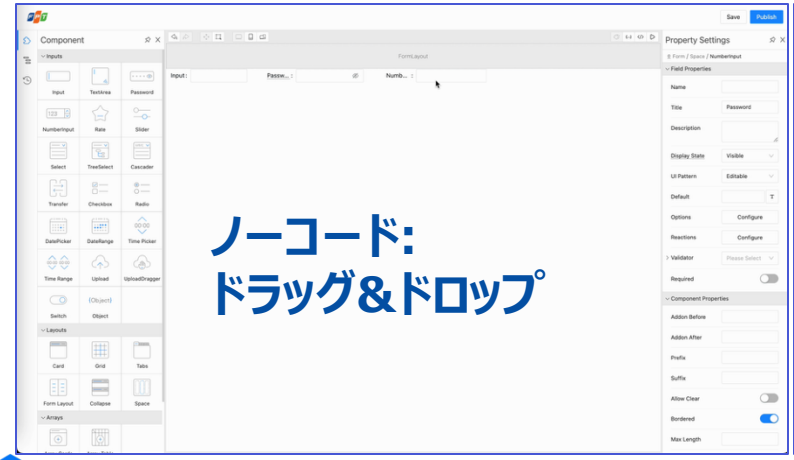
マルチレベルコーディングをサポート

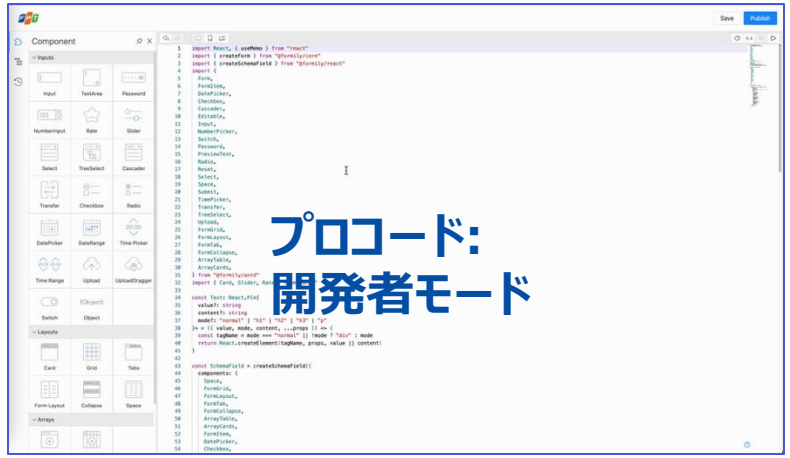


ユースケースサンプル適用

役割	タスク	□プラットフォームの実用例
ビジネスユーザー	ニーズの定義、フローのテスト	ノーコードのビジュアルビルダー、ドラッグ&ドロップUI
ソリューション設計者	ロジックの設計、ビジネスルールの検証	ローコード構成、JSONスキーマのカスタマイズ
開発者	機能の構築、システムの統合と拡張	フルコードアクセス、フレームワーク、API拡張

→ 1つのプラットフォーム、複数のペルソナ、統一されたワークフロー。

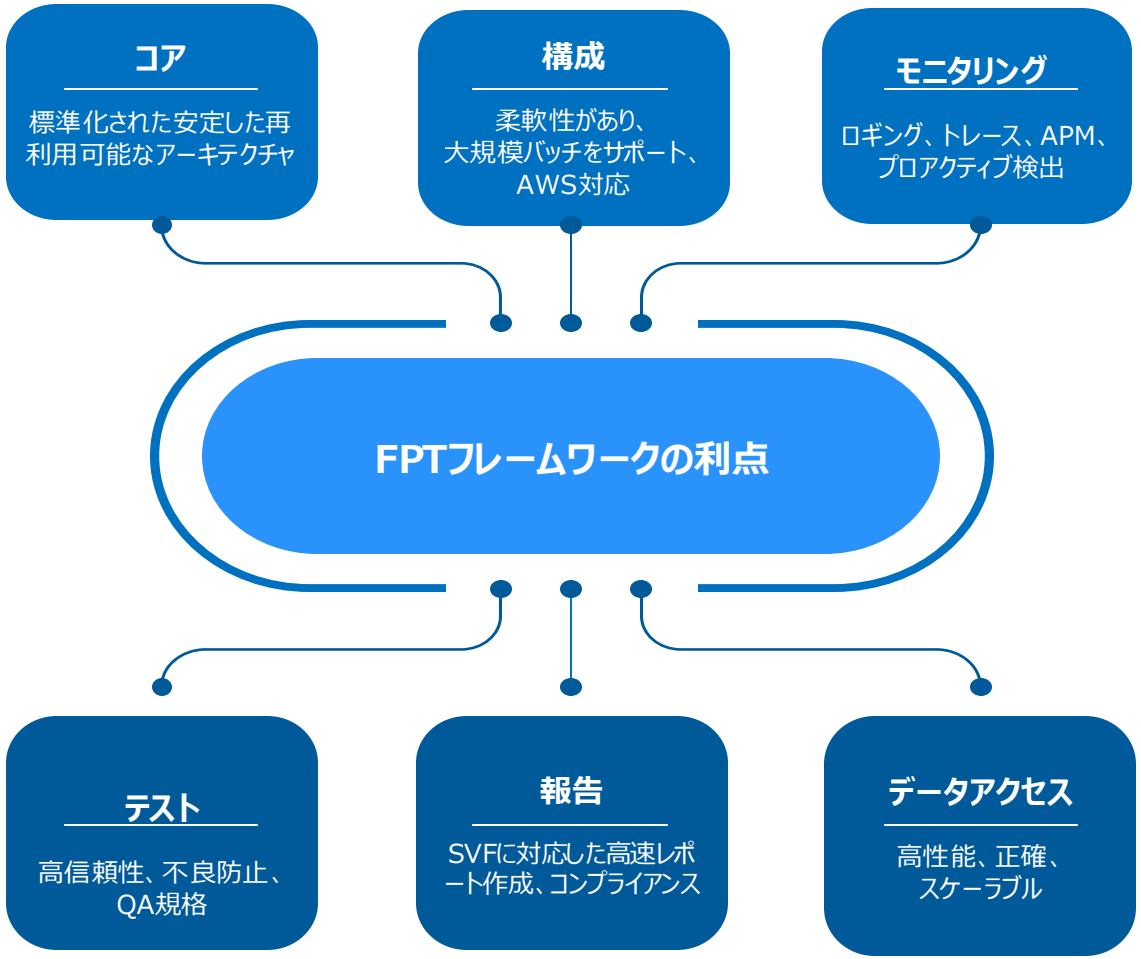


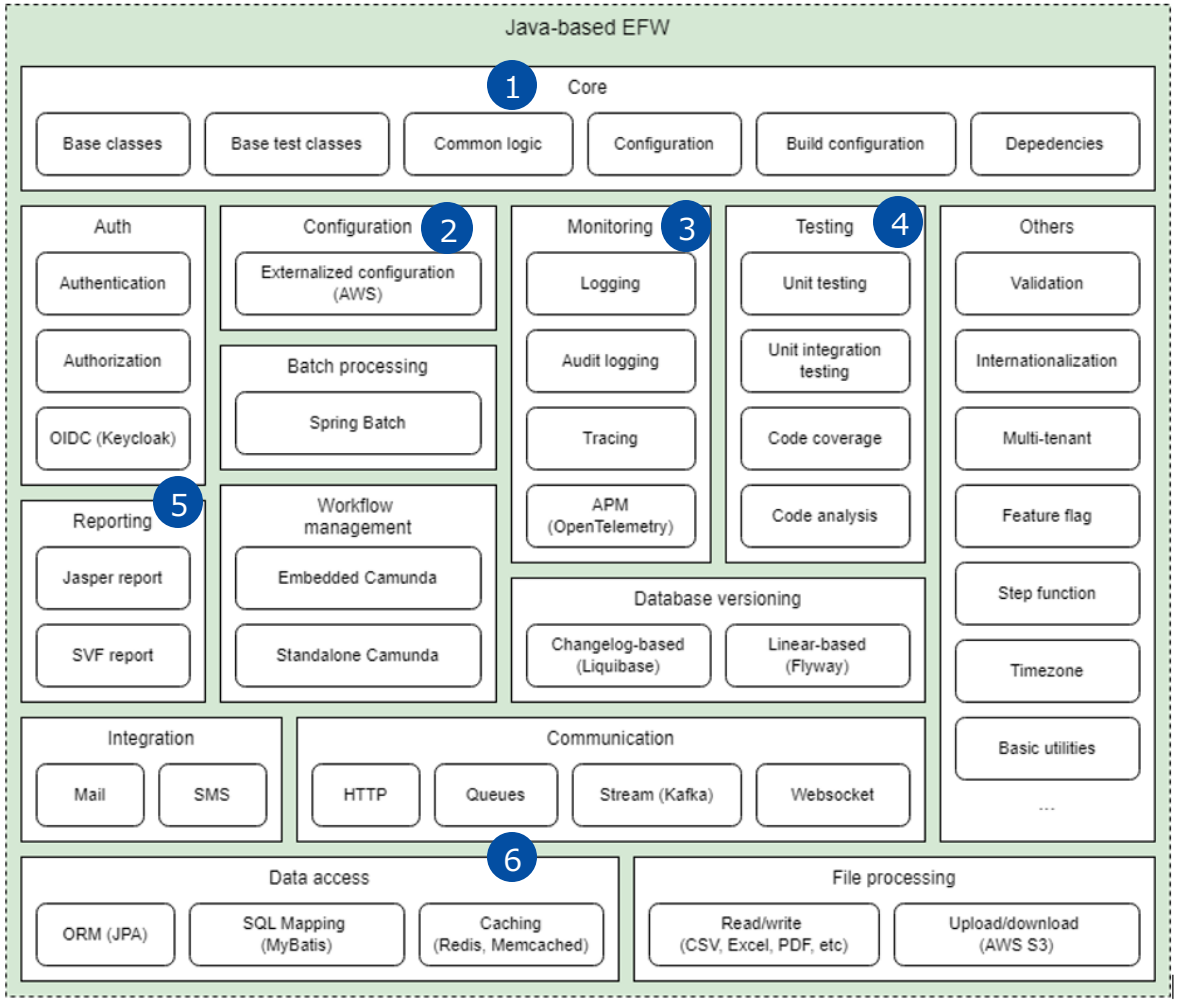


API化対応のバックエンドフレームワーク

この包括的なフレームワークは、開発者が高度でAPI対応のバックエンドシステムを効率的に構築できるよう、

ツールとコンポーネントのフルスイートを提供することを目的としています。





エンドツーエンドのサービスオファー

メインフレームのメンテナンス

- レガシー言語のメンテナンス
- 周辺アプリケーション開発

マネージドサービス

- アプリケーション・マネージド・サービス
- インフラストラクチャ・マネージド・サービス
- クラウドおよびハイブリッドクラウドマネージドサービス

品質検証とテスト

- システムテスト
- 統合テスト
- データ移行テスト
- 動作テスト

アセスメントとソリューション設計

- ディスカバリー&アセスメント
- ソリューション設計
- モダナイゼーション戦略/ロードマップ
- ビジネス・プロセス・リエンジニアリング

レガシーマイグレーション

- バージョンアップ
- オープンメインフレームへの移行
- リバースエンジニアリング
- AI-Baseソースコード変換
- データ移行

レガシーモダナイゼーション

- システムの再構築/書き換え
- APIfication/ API統合
- UI/UXのモダナイゼーション
- ビジネスのモダナイゼーション
- データ/データベースのモダナイゼーション

2,500+ COBOL、RPG、PL/Iエンジニア

4,500+ クラウドエンジニア

1,500+ データ&AIエンジニア

200+ SME
(自動車、BFSI、ヘルスケア、通信、製造など)

FPT EMT
Legacy Switch2Cloud

日本屈指のトップクラスの
鉄鋼メーカー

著名な日本の
不動産会社

EMT とGenAIテクノロジーを活用して、次を実現

- **600万行のコードを**統合
- リバースエンジニアリングタスクにおける人的労力を、早期評価フェーズで**30%**削減

EMTを使用して**データ抽出**と**レポート生成**を自動化

- フォーマットされていない入力ファイルに関する課題に対処
- 処理時間を**数か月から数週間に**短縮
- 大幅なコスト削減を実現

40+ のお客様

300+ システムの近代化

200+ MLOC 移行

サクセスストーリー：メインフレームのアセスメントとオープンシステムへの移行

ビジネスニーズ

2001年8月から稼働しているレガシー・メインフレームシステムは、技術の老朽化、システムの肥大化、複雑性といった課題に直面していました。これらの問題により、クライアントはビジネスの変化に迅速に対応することが困難となっていました。本取り組みは、システムを評価し、オープンでクラウドベースのプラットフォームへ移行することにより、機敏性の向上とコスト削減を目指したものです。

FPTのソリューション

- **準備・アセスメント**
  プロセスとツールの整備、資産の棚卸、WBSおよび現状（AS-IS）分析を通じて移行範囲を明確化
- **アーキテクチャ設計・PoC**
  TO-BEアーキテクチャを設計し、アプリケーションプラットフォームおよびインターフェースをカスタマイズ。PoCにより実現可能性を検証
- **コードの移行・比較テスト**
  ソースコードをJavaやShellなどのモダン技術スタックへ移行し、機能同等性を比較テストで確認
- **テスト（ITA、UAT、システム）**
  アプリケーションおよびハードウェア統合を含む、結合テスト、受入テスト、システムテストを実施
- **並行稼働と移行**
  デュアルシステムによるテストを行い、その後データおよび本番環境の移行を実施
- **移行・定着支援（ハイパーケア）**
  ユーザートレーニングとドキュメントを提供し、本番稼働後はモニタリングとサポート体制を強化

結果と効果

100% **資産可視化**

15% **解析時間の**短縮

3倍 移行分析の迅速な対応

0 **重大な障害**

サクセスストーリー：従来のメインフレームをMicrosoft Azureにモダナイズ

ビジネスニーズ

世界的な技術環境の進化に伴い、より持続可能な技術アプローチが登場し、クライアントがデジタルチャネルにおける成長機会を捉えることが可能になりました。クライアントは、IBMメインフレームをオープンテクノロジーへ移行し、既存のAzureクラウド上に展開することで、レガシーデータをBIやアナリティクスに活用し、新サービスや新たなビジネス機会に対する意思決定を迅速化したいとお考えです。同時に、多数のアプリケーションを効率的に管理し、保守コストの削減も目指しています。

FPTのソリューション

- **フェーズ1A – POC-1：** Roll Inventoryシステム（COBOL/DB2）をAzure上のJava/SQL Serverに移行し、1:1の機能互換性を実現
- **フェーズ1B – パイロット&リファクタリング：** UI/UXを再設計し、アーキテクチャをリファクタリングすることで、ユーザー体験と保守性を向上
- **フェーズ1C – 詳細調査&ロードマップ策定：** 技術および業務アセスメント、システムの詳細分析、アーキテクチャ定義を実施し、包括的なモダナイゼーションロードマップを作成

フェーズ2 – 実装

- 「安定化 → 機会創出 → 成長」の段階的アプローチを適用
- 移行済みのJavaコードを再利用し、再開発工数を削減
- Microsoft Active Directoryと統合し、アクセス管理を効率化
- Microsoft Azure上にアプリケーションを展開し、スケーラビリティとコスト管理を最適化
- FPTはAstadiaとの連携によりFastTrack Migration Suiteを活用し、自動化とトレーサビリティを確保
- IT部門と業務部門の関係者との連携を図るため、約20回のワークショップを実施

結果と効果

120.83% 生産性

0.24 人月あたりの**欠陥数**

20% **コスト削減**

削減 **手動工数**

サクセスストーリー：金融システム変革のためのRPGモダナイゼーション

ビジネスニーズ

- 老朽化したインフラ、高額な保守コスト、拡張性の制限といったレガシーシステムの長年の課題を解消
- ビジュアルデザインの強化、機能の充実、ユーザー体験の向上を備えた最新のクラウドプラットフォームへ移行
- プロジェクト開始から3年以内に、新しい本稼働システムをデリバリー

FPTのソリューション

レガシーAS/400システムが抱える高額な保守コスト、老朽化したインフラ、拡張性の制限といった課題に対応するため、クライアントはFPTと提携し、同システムをユーザーフレンドリーでクラウドネイティブなプラットフォームへとモダナイズしました。
このモダナイゼーションは3年間にわたり、2つの主要フェーズで実行されました。

パイロットフェーズ(2024年)

- RPGからJavaへの自動変換ツールを用いて、サブシステムSS9を移行し、実現可能性を検証
- 共通フレームワークを開発し、選定された機能を強化

本開発フェーズ（2024年～2026年）

- 残りの8つのサブシステムを移行・モダナイズ
- 新機能の統合および外部システムとのインターフェースを開発
- 品質保証、ドキュメンテーション、テスト自動化においてベストプラクティスとツールを適用

結果と効果

70% メンテナンス削減

2倍 開発スピード

クラウドイネーブルメント
完全にクラウドにデプロイされ、**スケーラビリティと最新のUI/UXを実現**

モダナイゼーションは**不可避**

成功のレシピは、ここにあります

クライアントの**信頼感**を
高める

計画を**現実的に**

ビジネスの**アジリティ**を
高める

運用を**効率化**

FPT EMT
Legacy Switch2Cloud

ありがとうございます！