

DXIT Forum Open Seminar

レガシー刷新と 攻めの変革の両立を目指して


2023年2月22日

東京システムハウス株式会社

独立行政法人 情報処理推進機構

比毛 寛之

溝口 則行



レガシー刷新と 攻めの変革の両立を 目指して（後半）

2023年2月22日（水）

自己紹介

比毛 寛之

ひもう ひろゆき



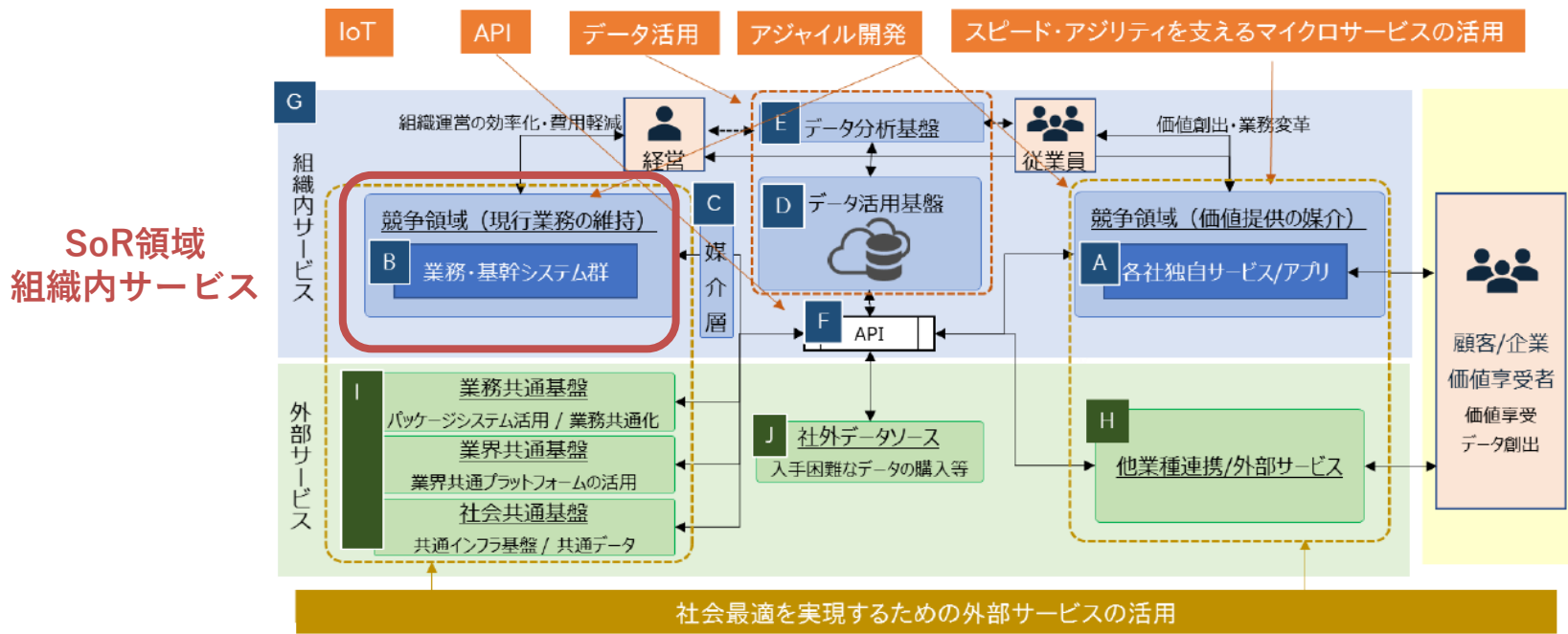
東京システムハウス株式会社
マイグレーションソリューション部 部長

- レガシーマイグレーション屋さん。COBOL資産のモダナイズや基幹系でのOSSの活用など、既存資産を活用した基幹系システムの刷新が得意。
- 活動：OSSコンソーシアム(理事)、COBOLコンソーシアム(세미나分科会)
- 出身：宮城県石巻市

モダナイゼーションの際に気になる事項

はじめに

- 本セッションの モダナイゼーション の対象



はじめに

• 本セッションのモダナイゼーションの手段

	BPR	リビルド	リライト	リホスト
システム構成要素	過去に縛られない ゼロからの開発	IT全体最適 業務処理最化	開發生産性向上 開発コスト低減	保守コスト削減 脱レガシー
運用	見直す	変更しない	変更しない	変更しない
情報	変更しない	変更しない	変更しない	変更しない
アプリケーション	運用に合わせて 再設計	現行仕様から 再設計	ソース書換 または変換	原則 変更しない
ミドル製品	オープン系製品	オープン系製品	オープン系製品	リホスト専用製品
OS	オープン系OS			
ハードウェア	オープン系サーバ（オンプレ、クラウド）			

資産再活用

はじめに

- 本セッションのモダナイゼーションの流れ



モダナイゼーションの際に気になる事項

- プラットフォームデジタル化指標(PFD指標)の評価項目より

対象	種別	大分類	項目数	
ITシステム全体	属性情報	財務	5	
		機能システム間の独立性		
	評価項目	データ活用の仕組み	12	
		運用の標準化		
		ガバナンス		
		プロジェクトマネジメント、品質 セキュリティ、プライバシー CIO、デジタル人材		
機能システムごと	属性情報	事業特性	13	
		影響度		
		システム特性		
		保有リソース		
		IT資産の状況		
	評価項目	①DX対応に求められる要件	データ活用性 ①	46
			アジリティ(ユーザ要件への対応) ②	
			アジリティ(非機能要件への対応)	
			スピード	
			利用品質	
②基礎的な要件	ITシステム品質	開発品質		
	IT資産の健全性 ③			

PFデジタル化指標は大きく分けて
ITシステム全体と機能システムごとから構成される。
さらにこれらは属性情報と評価項目から構成される。

本日取り上げる項目

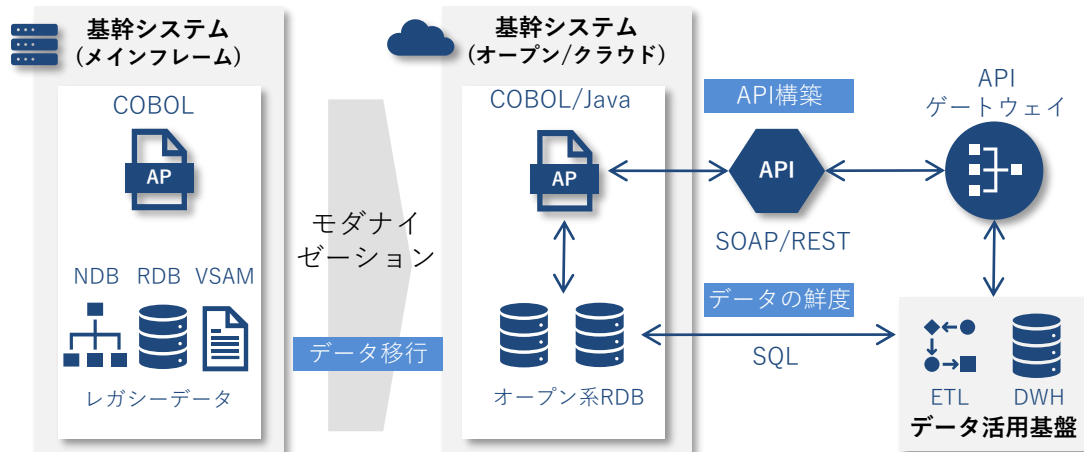
① データ活用性

指標の分類・項目	評価項目の説明	モダナイゼーション観点で気になる点
機能システムごと／評価項目(DX対応に求められる要件)		
データ活用性 ※SoR/SoEともに		
データの鮮度	活用すべきデータをリアルタイムに取得できるか。 ※リアルタイムに取得するデータは、リアルタイムにも、日次や月次などにも活用可能	DXのキモのひとつはデータ活用。そのためには活用できるようにする「仕組み」も必要で、従来は特定アプリだけが使っていたデータを活用できるかどうか。そのようなオープンさを持たせられるかどうか、モダナイゼーションへの期待のひとつではないか。しかし、そこまで期待をするのは無茶ぶりだろうか。
データの量の変化への対応	定義済データ項目について、必要十分なだけのデータ量を取得しているか。また、データ量を柔軟に拡張可能か。	
データ分析への入力方法	取得データを、AI（機械学習／深層学習など）や、データ分析のシステムに容易にインプットできる仕組みになっているか	

① データ活用性

回答

- ・無茶ぶりではありません。オープン系技術でモダナイズすることで、基幹系データのリアルタイム提供やAPIを通じた柔軟な提供ができます。
- ・データの鮮度は、論理構造や排他条件の精査、締め前提の業務仕様など、新鮮であれば良いのかについて十分な検討が必要です。



・データ移行

- ・レガシーデータをオープン系RDBに移行
- ・活用し易い列の検討（階層キー、マルチレイアウト、集団項目、OCCURS）

・API構築

- ・モノリスへの媒介層として、OLTP構築と同じ方法で元COBOLへのAPIを構築

・データの鮮度

- ・リアルタイムのデータ提供が可能だが、論理構造、排他条件、締め処理など新鮮さ vs 業務仕様を十分に検討

② アジリティ (ユーザ要件への対応)

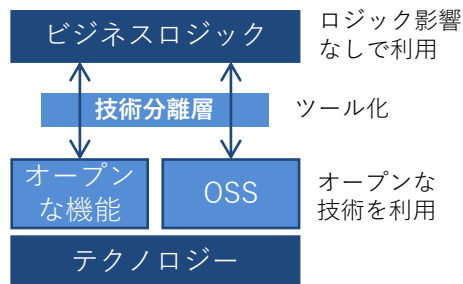
指標の分類・項目	評価項目の説明	モダナイゼーション観点で気になる点
機能システムごと／評価項目(DX対応に求められる要件)		
アジリティ(ユーザ要件への対応)		
要件変更し易い 実装	小さい業務機能などの単位で、独立して開発できるような作りになっているか(適切なデータ分離、機能分割、構造化、カプセル化、重複や矛盾のないデータ、必要十分な設計情報の記述、など)	外部エコシステムと連携するには、旧システムとは異なるシステム間連携やミドルウェア(比較的新しい技術)に対応したい場合もあるだろう。その場合、採用したモダナイゼーション手法やツールが事前に想定されているものでないと対応困難と推察する。希望する連携を実現するならば結局は新規開発をせざるを得ないだろうか。
迅速な対応のための組織・体制	事業責任者、業務担当、システム担当が三位一体となって、製品/サービスのシステム対応を迅速に実施できる組織・体制を取っているか ※システム対応の度に、社内調整、承認などで時間をかけないため	
エコシステムの活用、連携の容易さ	外部のエコシステムの活用・連携が容易な方式を取っているか 例:サーバレスなビジネスロジック実行基盤、NoSQLデータベース	

② アジリティ (ユーザ要件への対応)

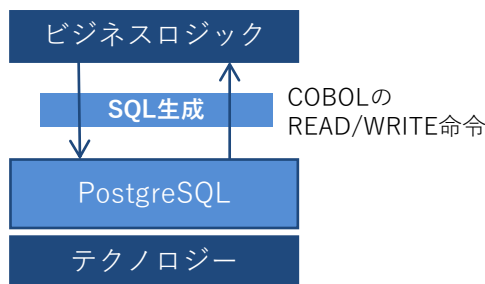
回答

- ・新ミドルや新技術ではテクノロジーとビジネスロジックの分離が重要です。COBOL再活用の場合は、COBOL命令のまま利用できるようにします。
- ・個別対応をしないようにツール化し、開発者が向き合うビジネスロジックには影響しません。将来、テクノロジーを変更する際の修正を最小限にします。

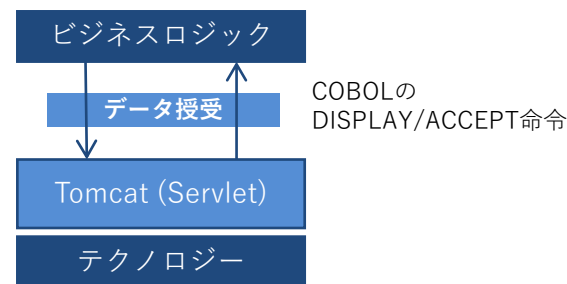
<基本方針>



<対応例1>



<対応例2>



テクノロジーの変更は、技術分離層を変更すればOK、ビジネスロジックの変更は最小限外部のエコシステムと連携する場合も同様の方式を取る

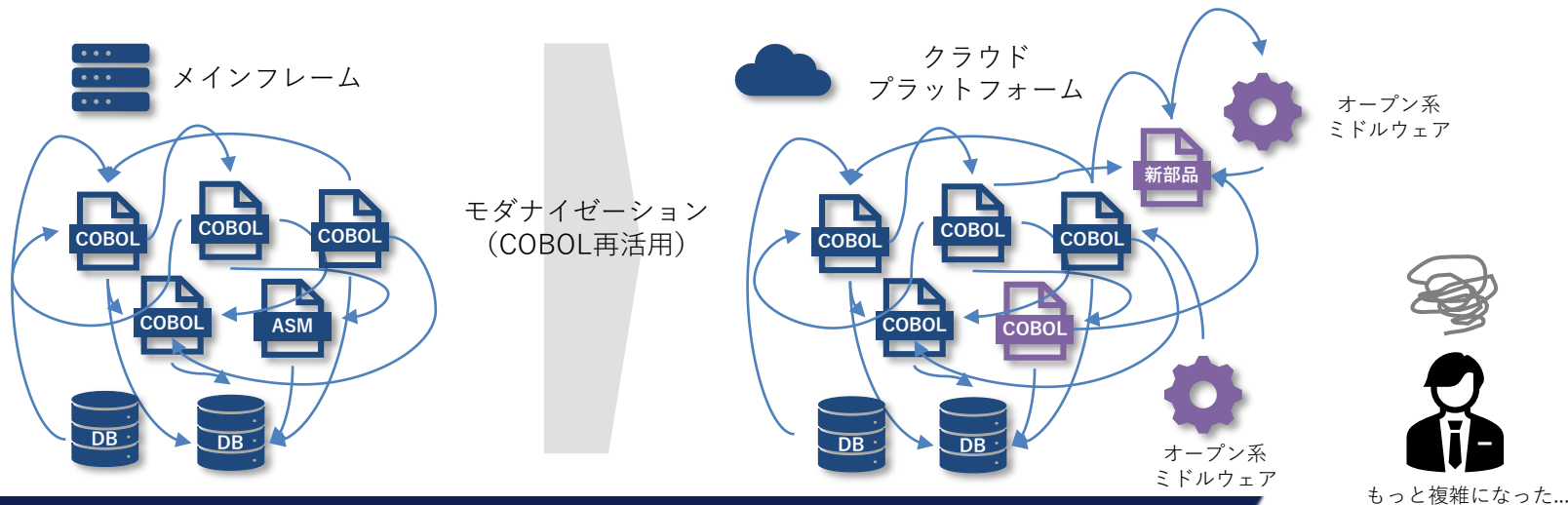
③IT資産の健全性

指標の分類・項目	評価項目の説明	モダナイゼーション観点で気になる点
機能システムごと／評価項目(基礎的な要件)		
IT資産の健全性		
ソフトウェア資産の最適化	<p>品質管理標準に従って、最適な設計・ロジックの追加や修正が継続的に実施された結果、ソフトウェア資産は最適な状態になっているか。</p> <p>※目的:ロジックを簡潔にして、スパゲッティ化を防ぐため</p> <p>※問題となる例: 難解で、修正しにくい、デグレし易い、コーディング不適切なモジュール構成・分割</p>	<p>元のアプリが複雑化・スパゲティ化している場合は、そのまま移行することにならないだろうか。同様に、使われないコード部分もそっくり移すことにならないか。</p> <p>保守性・変更可能性を確保するために、ソフトウェア資産を最適化するための工夫はないだろうか。</p>
不要なソフトウェア資産を増やさない	<p>使われないコーディング、不要なコーディングを組み込まない、かつ、共通的な処理は部品化され、活用するよう徹底した結果、不要なソフトウェア資産がない状態になっているか</p> <p>※目的:ソフトウェア資産の肥大化を防ぐため</p> <p>※問題となる例:使われない、または無駄なコーディングがソースコードに含まれているあちこちに似たようなコーディングをしている機能が少ない割に規</p>	

③IT資産の健全性

回答

- はい、現行のスパゲッティ化・複雑化の状態はそのまま移行されます。そこにホスト機能代替やミドル連携が加わり、もっと複雑になります。
- システムが長年の手入れ不足の場合、IT資産の健全性を取り戻し、保守性や変更容易性の改善と再レガシー化しない方式を検討します。



③IT資産の健全性

ところで、IT資産にCOBOLを残すのか？

- 最近のCOBOL言語の話題
 - 基本情報処理技術者試験でCOBOLの出題終了
 - COBOLは古い、新しいことできない・・・
 - COBOLのシステムは複雑、やりたくない・・・
 - COBOL技術者の引退、技術者不足
 - 空前の脱COBOLブーム（Javaリライト）
 - 今もCOBOL資産は世界に8000億行ある
（Micro Focus社 2022年2月調査）



黙々と処理をこなして社会基盤を支える巨人COBOL。
その姿はギリシャ神話で天空を背負うアトラスのようだ。
COBOLが手を離せば天空が落ちてくる。
（画像 Wikipediaより）

COBOL言語の問題ではなく「IT資産の健全性」の問題

③IT資産の健全性

- 本来COBOLは保守性や変更容易性が高い言語
 - 国際規格で標準化、この1月に新規格公開（ISO/IEC 1989:2023） NEW
 - 英語を用いた分かり易い構文、事務処理計算の業務を抽象的に記述
 - 構造化プログラミング、コード再利用の仕組み
- 手入れ不足で不健全になってしまった
 - **長年の増改築**（コピペ量産、念のため残した処理、その場しのぎのバイパス処理）
 - **標準でない記述**（ベンダー独自機能、特定機器向け制御コード）
 - 仕様書や**テストケースの消失**（元々は綺麗に整っていたはず）
 - COBOL技術者不足よりも深刻な**仕様理解者不足**（ベテラン引退、引継ぎ不足）

COBOL再活用 でも 脱COBOL でも良いが、まず健全性の確保を

③IT資産の健全性

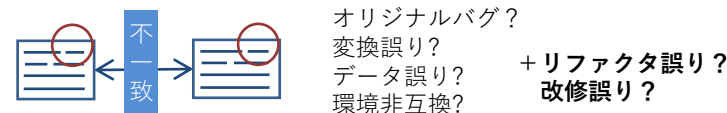
モダナイゼーションにおけるIT資産の健全性のポイントは4点

- 1 プロジェクト開始前のリファクタリング
- 2 テクノロジとビジネスロジックの分離（再レガシー回避）
- 3 テストケースの再構築
- 4 仕様理解者を増やす

- 無駄なコードは移行コストに直結する



- モダナイ中のリファクタリング・改修はNG。現新比較の不一致原因が増加してしまう。



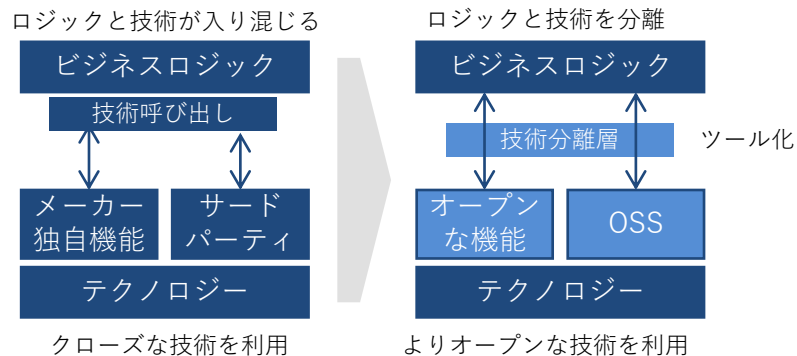
- 保守性・変更容易性の観点で対策
 - デッドコード、なぞ変数名、バイパス見直し
 - スパゲティは仕様理解者または解析ツール活用
 - 現行システム上でしっかり動作確認

③IT資産の健全性

モダナイゼーションにおけるIT資産の健全性のポイントは4点

- 1 プロジェクト開始前のリファクタリング
- 2 テクノロジとビジネスロジックの分離（再レガシー回避）
- 3 テストケースの再構築
- 4 仕様理解者を増やす

- 技術変化に左右されない設計
 - ビジネスロジックがテクノロジーを自由に乗り換えられる仕組みにする



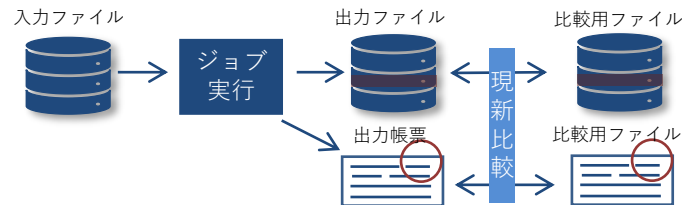
- 再レガシーの回避
 - メーカー独自機能を互換製品で回避すると将来の再レガシーの原因となり得る

③IT資産の健全性

モダナイゼーションにおけるIT資産の健全性のポイントは4点

- | | |
|---|-----------------------------|
| 1 | プロジェクト開始前のリファクタリング |
| 2 | テクノロジーとビジネスロジックの分離（再レガシー回避） |
| 3 | テストケースの再構築 |
| 4 | 仕様理解者を増やす |

- 現新比較テストはモダナイ後も維持すべき**
 - 構築当初のテストケースが手入れ不足 or 消失
 - モダナイは現新比較テストを必ず作る
 - 廃棄せず、せっかく作ったのだから維持をする



- CI/CDパイプラインで現新比較を再活用する**
 - バージョン管理 + CIツールを導入する
 - クラウドのAWS code pipeline、Github Actionsなどで基幹系向けのCIを構築する

③IT資産の健全性

モダナイゼーションにおけるIT資産の健全性のポイントは4点

1	プロジェクト開始前の リファクタリング
2	テクノロジーとビジネスロジック の分離（再レガシー回避）
3	テストケースの再構築
4	仕様理解者を増やす

- 仕様理解のチャンスをベンダに丸投げしない
 - 仕様理解とスキルセットへの投資と考える
 - スキルセットを取り戻し将来構想の足掛かりへ
- ブラックボックスを紐解く ⇒ 結果が明白
 - 分析設計：棚卸、機能整理、サードパーティ整理
 - 現新比較：シナリオ/データ作成、不一致の調査
 - 総合テスト：運用・保守のテスト、シナリオの作成、ジョブスケジュール、外部連携の検証
- その過程でベテランから若手へのノウハウ継承
 - 若手はシナリオ/データ作成で現行を理解、次第に不一致の詳細調査で深部の仕様も把握
 - 若手の方がオープン系技術が得意なので、終盤にかけてウエイトは若手に移っていく

まとめ

- **モダナイゼーションの際に気になる事項**
 - データ活用性
 - アジリティ（ユーザ要件への対応）
 - IT資産の健全性
- **モダナイゼーションに向けて**
 - PFD指標で現在システムを診断して適切な移行手法を検討
 - モダナイは数十年の歴史あるシステムのDXに向けた第一歩（未完結）
 - ベンダ任せにせずお客様自身がスキルセットを得て将来構想の足掛かりへ

[参考資料]

東京システムハウスのご紹介

会社概要

東京システムハウスのご紹介

- 商号** 東京システムハウス株式会社（略称：TSH）
- 設立** 1976年11月
- 資本金** 1億7,990万円（払込資本）
- 売上高** 30.0億円（2022年10月期）
- 従業員数** 174名（2022年10月現在）
- 本社** 東京都品川区西五反田8-1-5 五反田光和ビル5F
- 特色** 独自技術に特化したソリューション・サービス。
ニッチな業種向けの専用パッケージ。どの企業系列にも属さない独立系。
コンピュータ利用に関する総合サービスを提供。



マイグレーションの取り組み



1995年から蓄積された
経験・ノウハウと
230件以上の導入実績



代替フレームワーク
「AJTOOL」の開発と提供

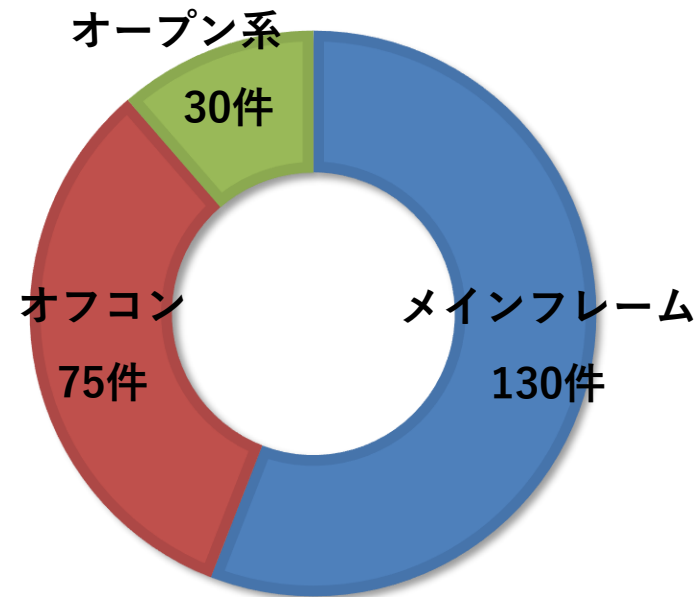


COBOLシステムの
クラウドネイティブ移行
に対応

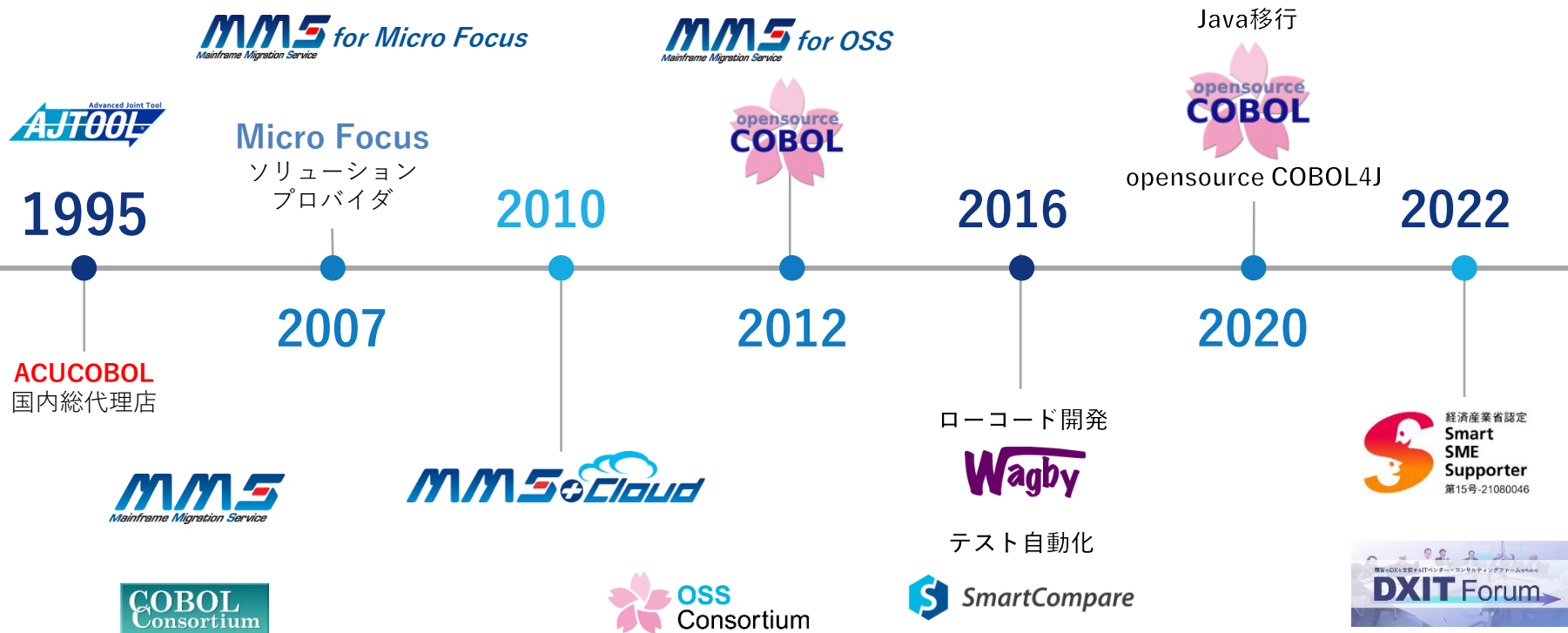


コミュニティでの
開発貢献と基幹系での
OSS利用の普及活動

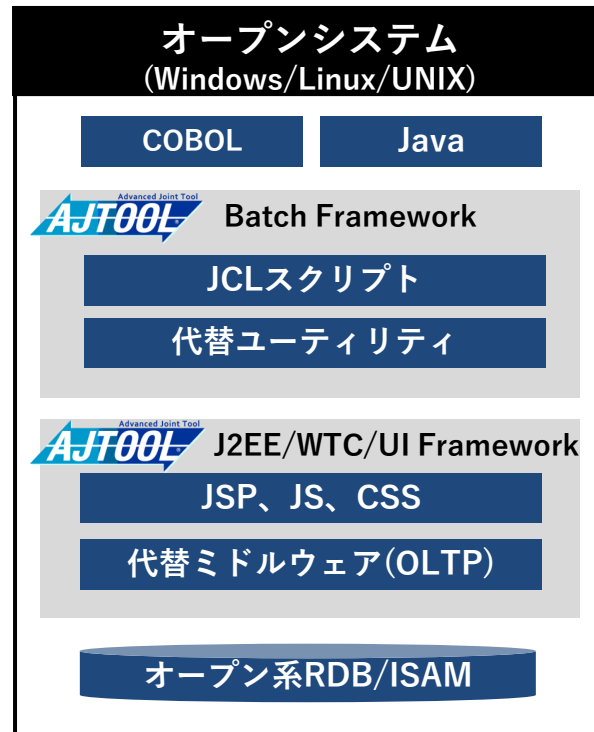
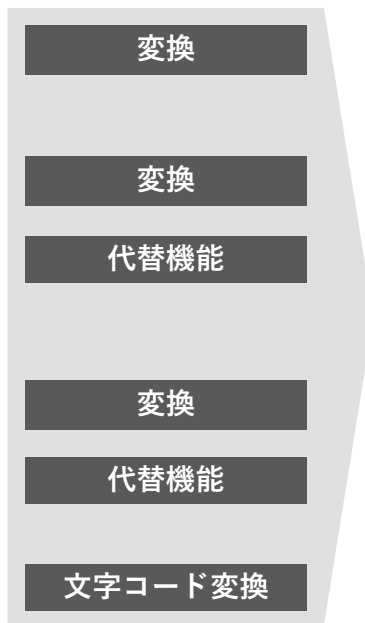
マイグレーション実績
(1995年～)



マイグレーションの取り組み

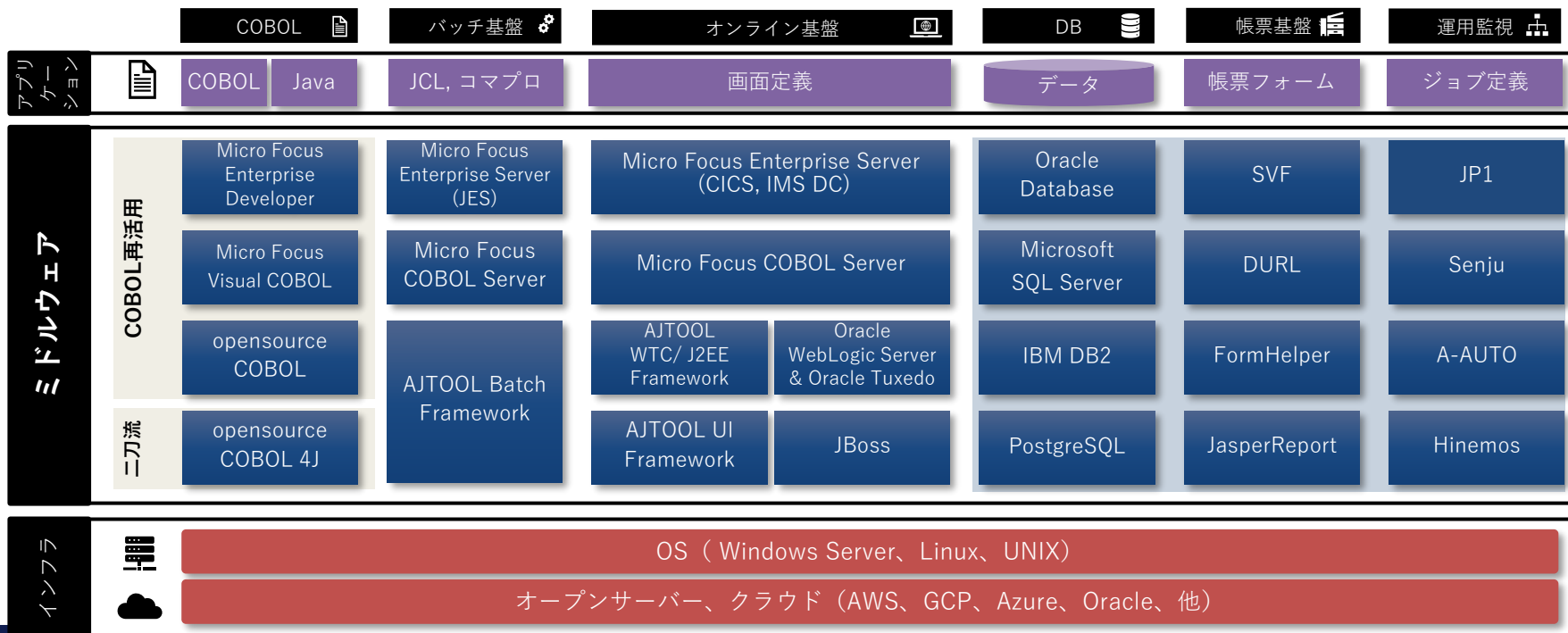


レガシーマイグレーション（資産移行）



レガシーマイグレーション（環境移行）

お客様に合わせた最適な処理方式とミドルウェアへ移行します。



マイグレーションの工程

分析・設計フェーズ

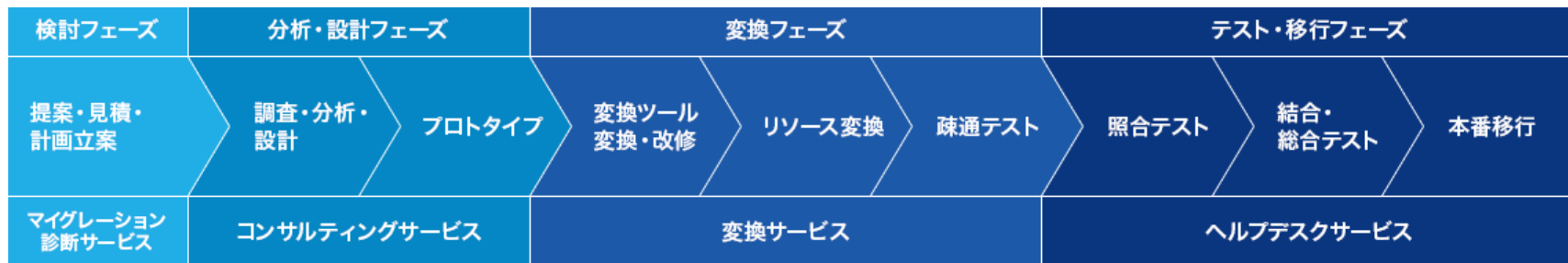
- ・資産分析から課題を抽出し、**網羅的解決**を検討する
- ・一部資産でプロトタイプを作成し、移行性や機能性および性能の検証を実施

変換フェーズ

- ・**変換ツールでの自動変換**や一部の手直しと、代替機能の製造や単体テストを実施
- ・疎通テストで変換結果の簡易検証を行う

テスト・移行フェーズ

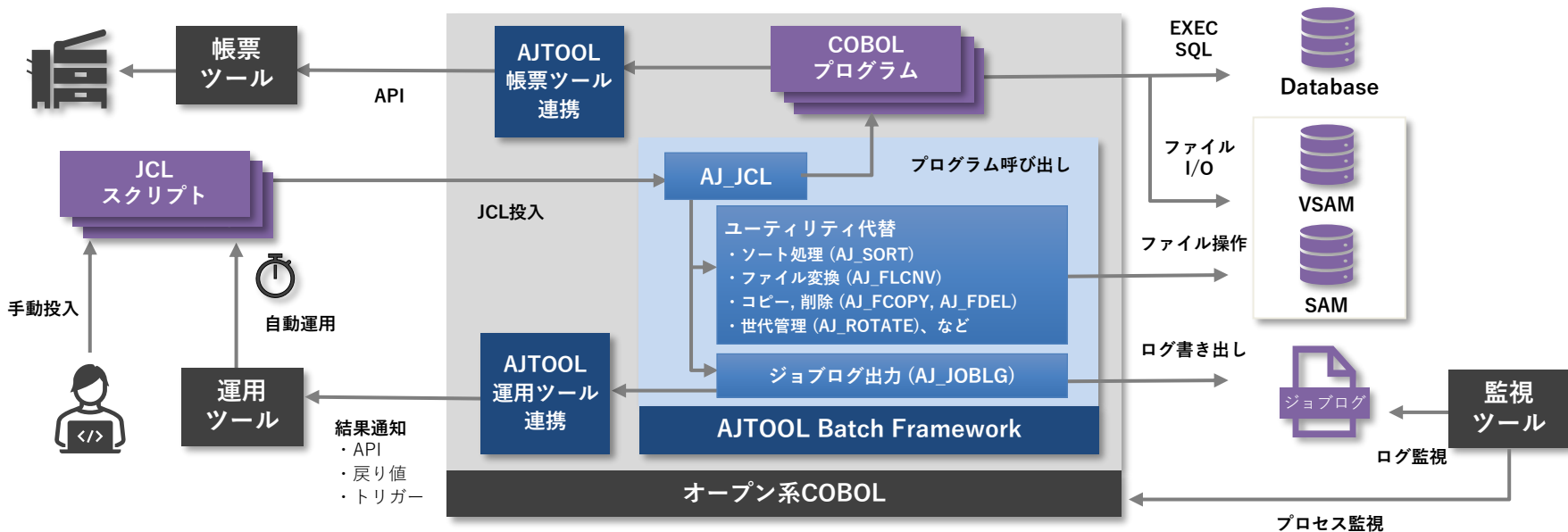
- ・**照合テストで現新比較**を行い、移行の正当性を確認
- ・**総合テスト**で運用を見据えた動作確認を実施



AJTOOL Batch Framework

バッチ処理の実行フレームワーク

- JCLスクリプトの実行制御 AJ_JCL とユーティリティの代替機能を提供します

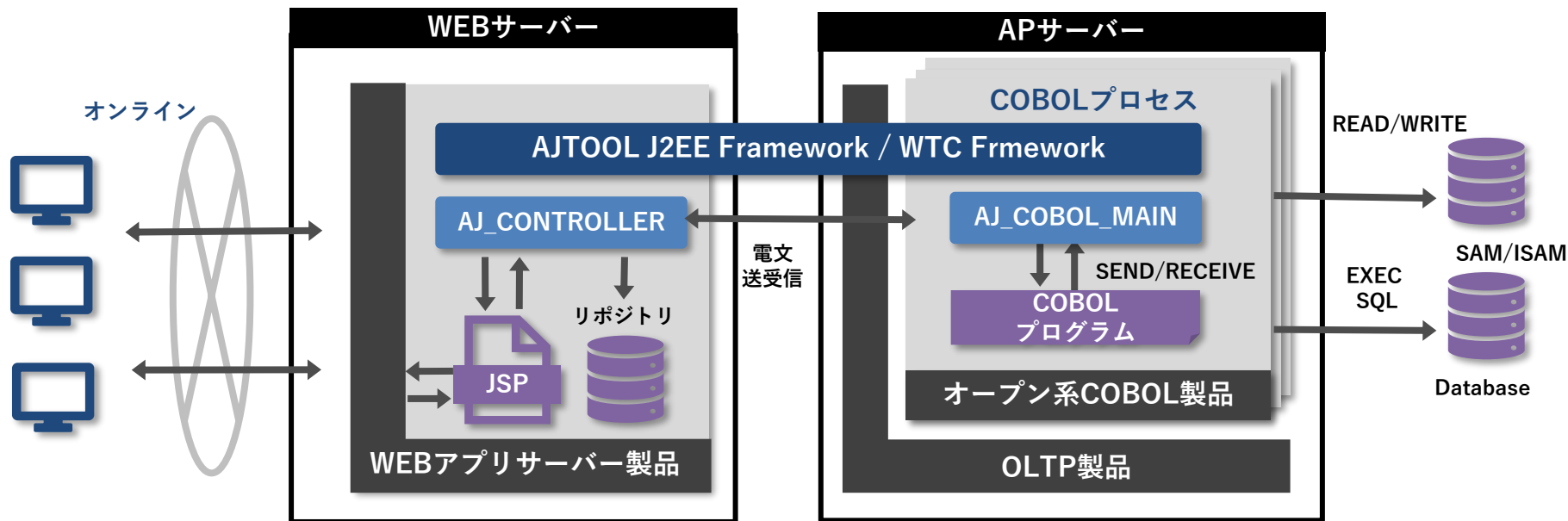


AJTOOL J2EE Framework / WTC Framework



COBOLオンライン処理のフレームワーク

- WEB⇒OLTP⇒COBOLが連動するオンライン処理を実行します

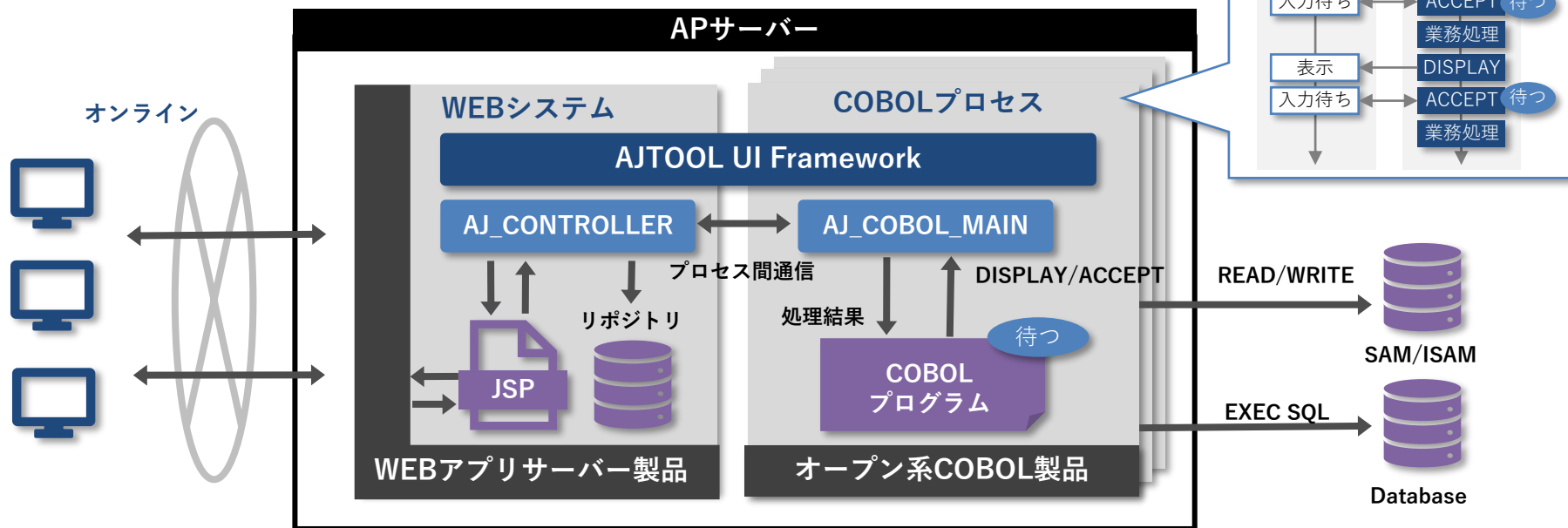


AJTOOL UI Framework

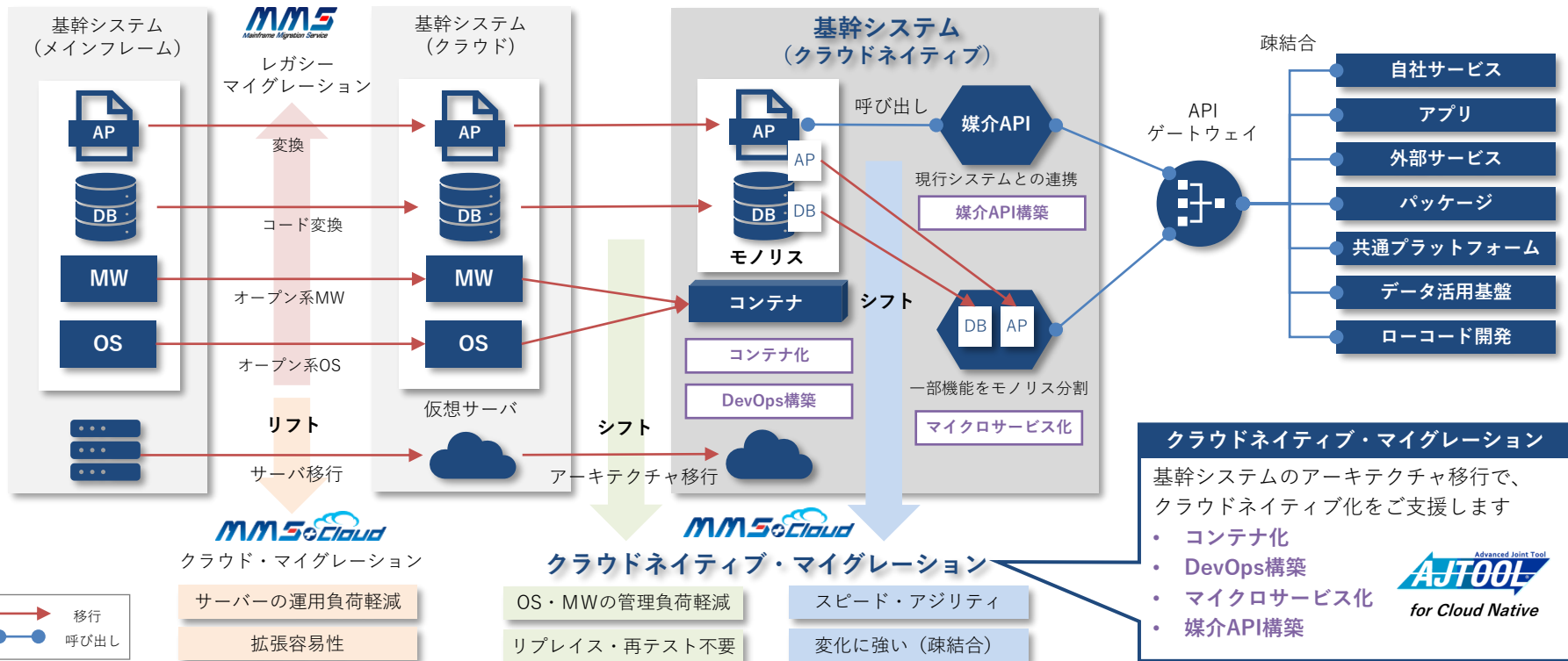


オフコン型COBOLのオンラインフレームワーク

- 業務処理と画面I/Oが連動するステートフル処理を再現します



クラウドネイティブ・マイグレーション



マイグレーション最新事例

• ITモダナイゼーションSummit 2022

- レビュー記事：

日経クロステック Special

『独自ツールの活用から“二刀流”まで顧客に合った“脱レガシー”を提供』

https://special.nikkeibp.co.jp/atclh/NXT/22/it_modernization_summit0531/p7/

- 講演資料：http://www.cobol.gr.jp/knowledge/material/220414_report/01.pdf

- 日本製紙様（富士通ホストからのマイグレーション事例）

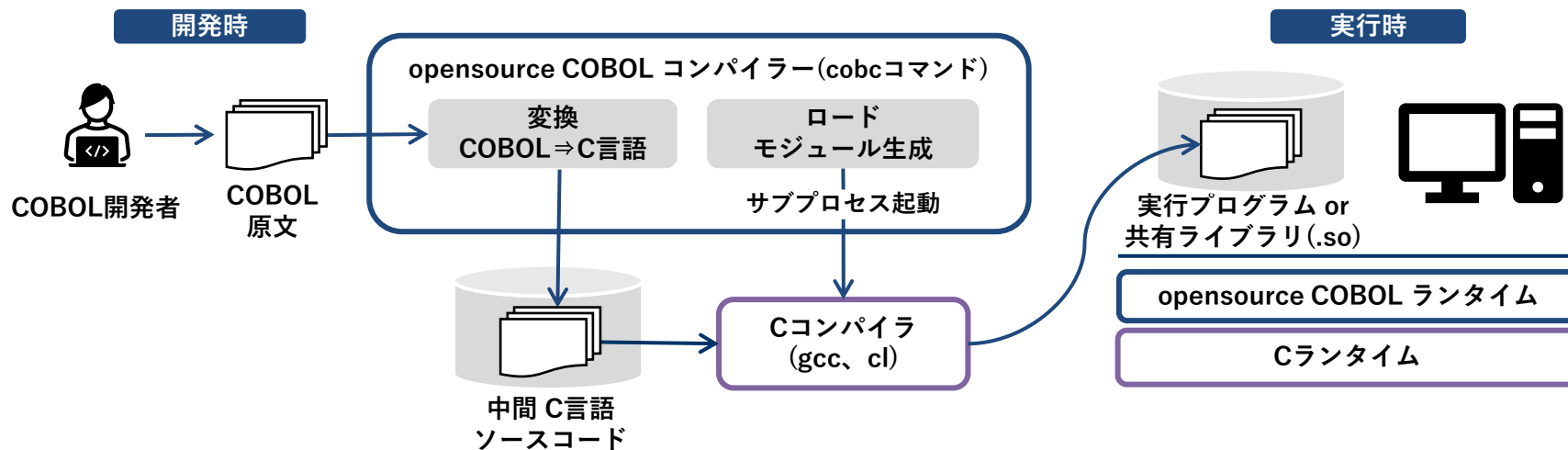
- 北陸電機商会様（東芝TPCareからのマイグレーション事例）

opensource COBOL

<https://github.com/opensourcecobol>



- OSSコンソーシアムで開発・公開しているCOBOLコンパイラ
- COBOLをC言語にトランスレート、Cコンパイラでバイナリを生成
- Linuxは『gcc』、Windowsは『cl』やLinuxエミュレータ(CygWin/MinGW) を利用

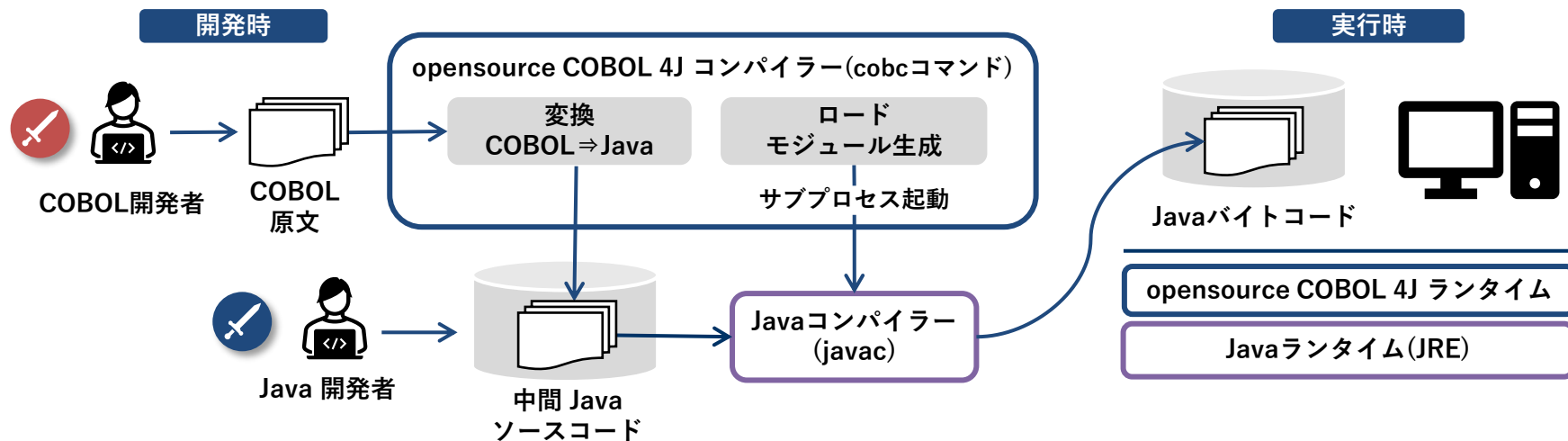


opensource COBOL 4J

<https://github.com/opensourcecobol>



- opensource COBOL シリーズの新しいプロジェクト (OSSコンソーシアム)
- COBOLをトランスレートして Javaを生成、javacでバイトコードを生成
- 元祖「opensource COBOL」を Java にマイグレーション



opensource COBOL 4J お試しく下さい

ダウンロードはこちら

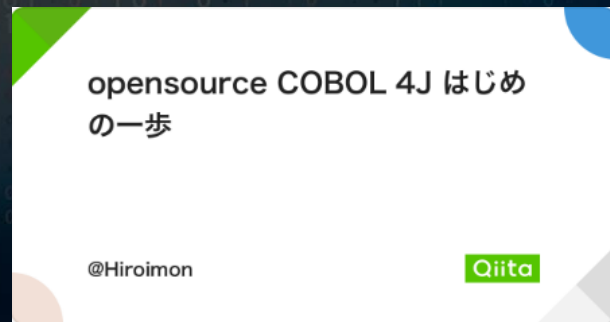
<https://github.com/opensourcecobol/opensourcecobol4j>

Qiita記事『openource COBOL4J はじめの一步』

<https://qiita.com/Hiroimon/items/a73e8bdec4b376916ca0>

最新情報はTwitterでも発信中

https://twitter.com/tsh_mms



まとめ

- 会社概要
- マイグレーションサービスの紹介
 - レガシーマイグレーション
 - クラウドネイティブ・マイグレーション
- opensource COBOLのご紹介
 - opensource COBOL
 - opensource COBOL 4J (Java方式)

お問い合わせ

東京システムハウス株式会社
マイグレーションソリューション部



☎ 03-3493-4601

✉ mms@tsh-world.co.jp

🐦 [@tsh_mms](https://twitter.com/tsh_mms)

本文に記載されている会社名、商品名は一般に各社の商標または登録商標です