

「COBOLと共にフロンティアを目指して」  
～COBOLクラウドサービスによるIT資産の有効活用のご提案～



NTTデータの総合クラウドサービス

**BizXaaS**

2011年4月15日

株式会社NTTデータ

星野 亨(エグゼクティブITアーキテクト)

## 本日の内容

- 1 レガシーシステムとは？
- 2 ユーザー部門の抱える問題
- 3 脱レガシーの課題
- 4 COBOLクラウドサービスとは？

**本セッション参加の方で自社でレガシーシステムをお持ちの方  
挙手をお願いします。**

現職：基盤システム事業本部 基盤サービスビジネスユニット

事業内容：

① ミッションクリティカルソリューション

**PORTOMICS**

NTTデータの総合クラウドサービス

**BizXaaS**

② レガシーマイグレーション「BizXaaSマイグレーションサービス」

'86	'87	'88	'89	'90	'91	'92	'93	'94	'95	'96	'97	'98	'99	'00	'01	'02	'03	'04	'05
BK-SMP	基幹システム DISK更改	基幹システム 更改	端末 設計	新システム 開発	基幹システム 更改	電子政府	PORTOMICS 開発												
ミドルソフト 設計・開発	移行設計	ミドルソフト 設計・開発		性能・設備設計	性能・設備設計	PM	PM												
メインフレーム (DIPS)			OS/2			メインフレーム (民需機)			オープン系										
DIPS-11/5E		DIPS-11/5EX	日立MP5800		富士通 GS8800		日立 MP5800							NEC PX7800					
DIPS: 電電公社開発メインフレーム Dendenkoshi Information Processing System			パラダイム シフト			パラダイム シフト													
パラダイムの異なる3つのプラットフォーム上での金融・官公庁を中心とした基幹システム開発																			



# レガシーシステムとは？

- 長年積み重ねてきた動作実績
- 専用ハード・ソフトによるベンダプロプライエタリ(垂直統合)なアーキテクチャ  
そのため統一感があり、運用がしやすい(使いやすい)
- 単一ベンダのため、故障時対応の迅速性、原因究明、責任範囲が明確であり、利用者にとっては楽

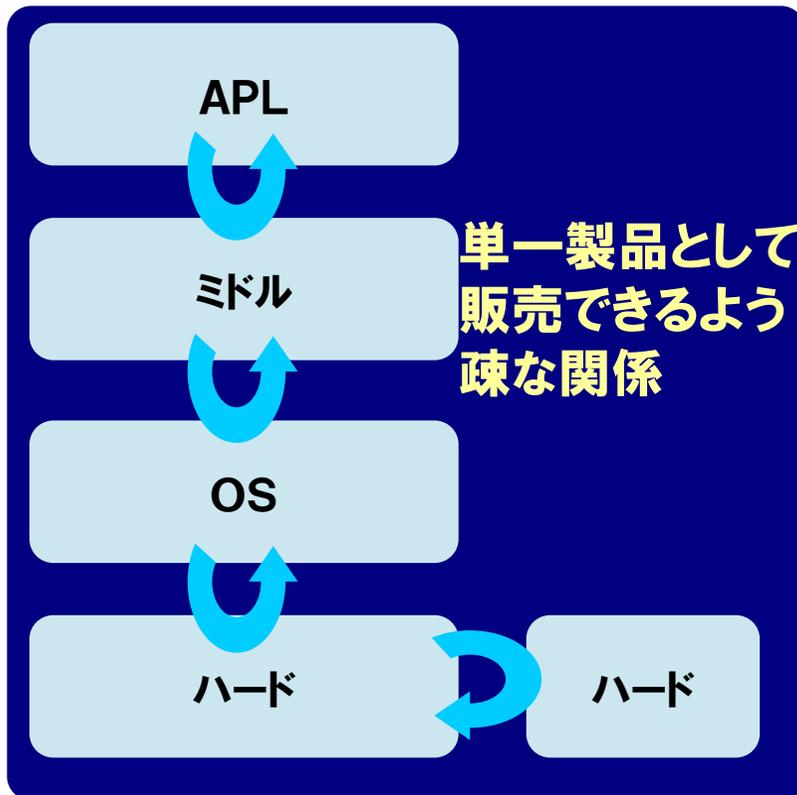


	メインフレーム	オープン系
アーキテクチャ	インテグラル(統合型)	モジュラー(部品型)
インタフェース	独自(call-return、非同期通知)	標準(call-return)
設計思想	単一ベンダ製品、すりあわせ	製品選択の自由、疎結合(BlackBox)

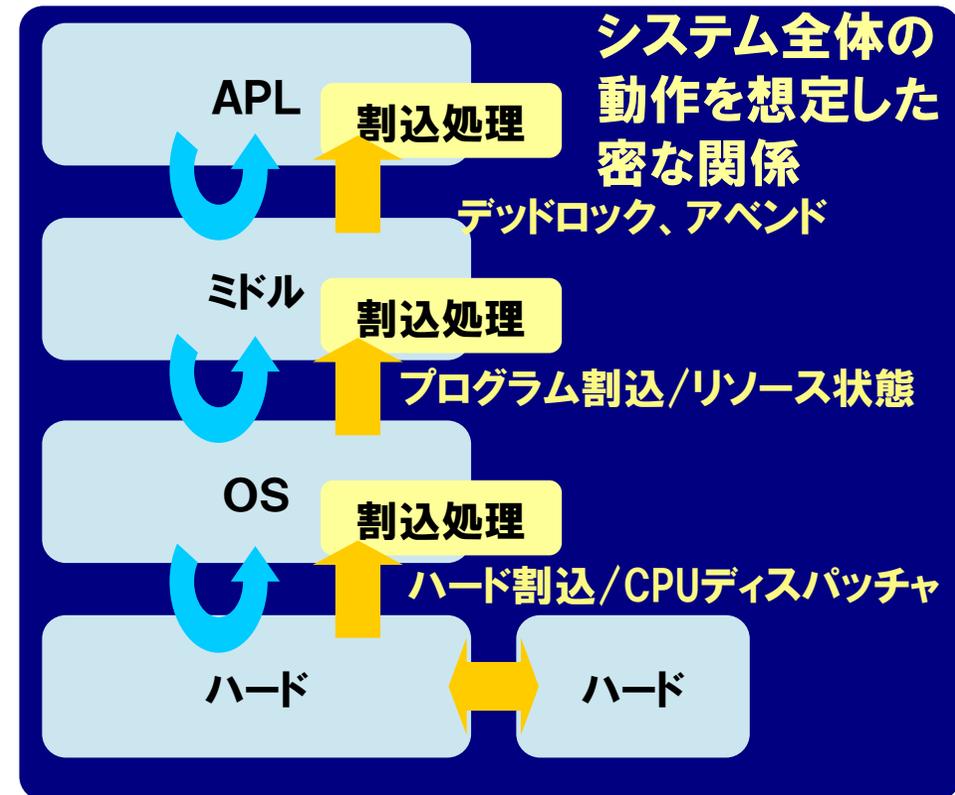
# モジュラー型とインテグラル型アーキテクチャの違い

「モジュラー型」 標準化により、製品相互の組み合わせ自由度がある反面、異常時等を想定したシステムインテグレーションが必要。

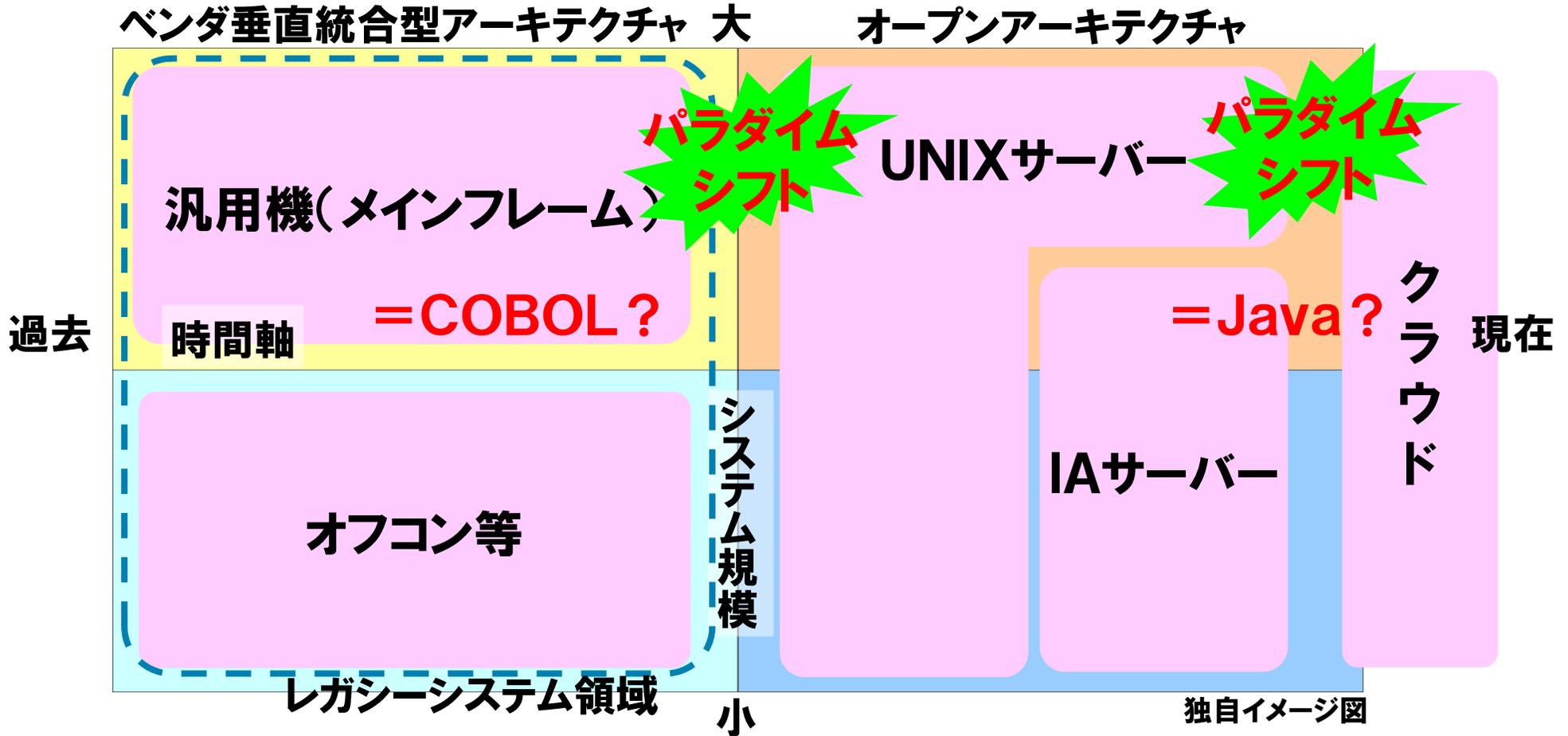
「インテグラル型」 製品組み合わせを限定し、製品間のすり合せによるきめ細かい連携が実現できる。



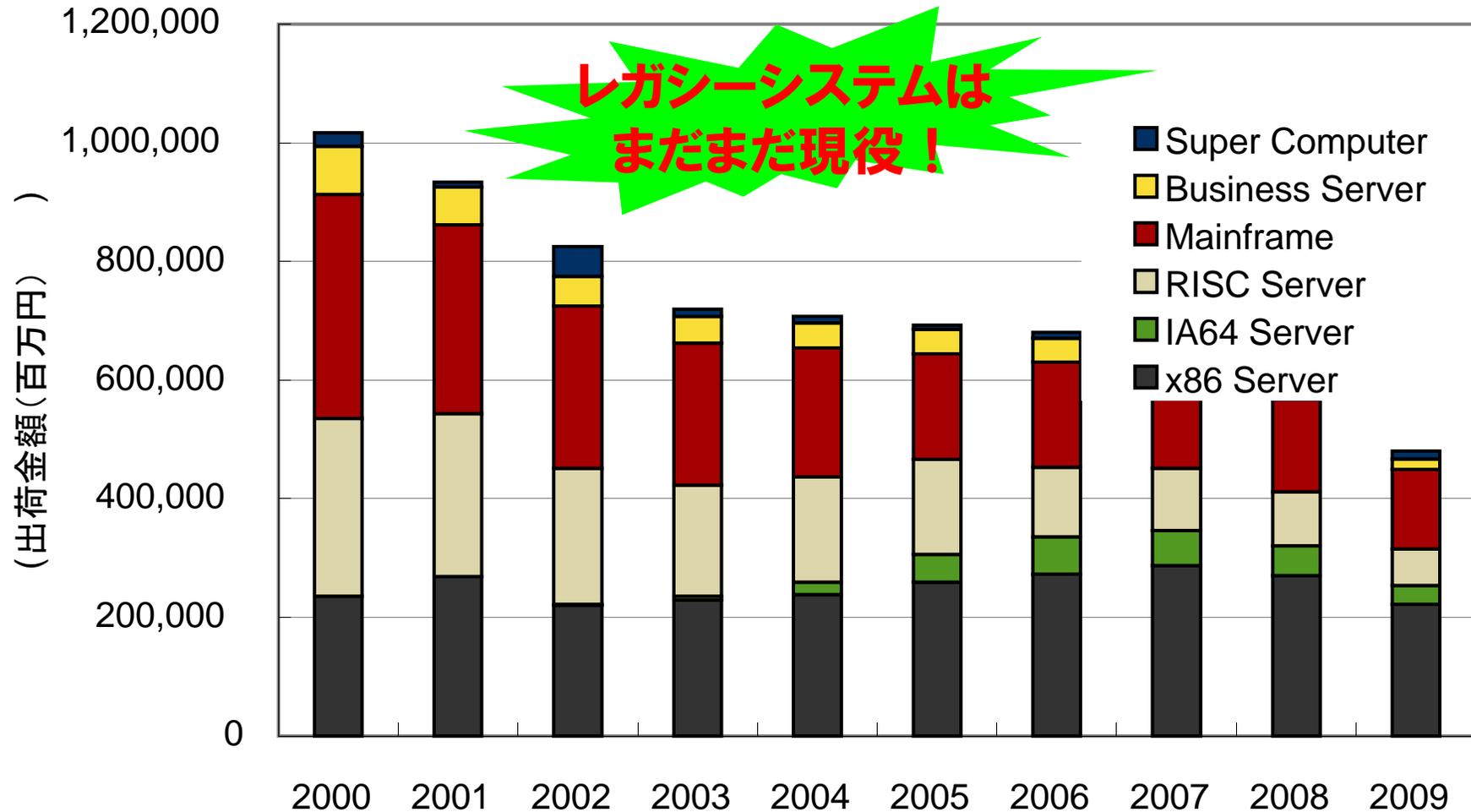
モジュラー型アーキテクチャ



インテグラル型アーキテクチャ



# レガシーシステムの現状



出典元: IDC Japan (2010)

## レガシーシステムにも二種類ある

### Legendary(伝説的)なシステム

重要度が極めて高く、サービス継続性・性能、処理の一貫性を求められるシステム  
(銀行勘定系や社会インフラ系システム)【Mission Criticalシステム】

メインフレームの  
継続活用

### Legacy(遺産・遺物)なシステム

メインフレームしか選択肢の無かった時代からのシステムでこれまで更改や再構築の  
機会がなかった処理【Non Coreシステム】

モダナイゼーションに  
よるTCO削減



高

重要基幹システム

レガシー継続利用

インフラシステム

BPR・リビルドによる  
ROIの追求

リホストによるTCO  
削減

求められるSLA

大

システムの戦略性

小

Java等へのリビルド  
による生産性向上

リホストまたは  
パッケージの活用

一般システム

ノンコアシステム

低

独自イメージ図

Common Business Oriented Language（汎商業目的言語）

1960年に最初の仕様書が発行され、現在まで業務処理言語として活用

## 【ソース資産】

蓄積されたプログラム資産は稼働中だけで全世界で約2000億行とも...

世界の民間企業の8割(2000年時点)がCOBOLプログラム

日本国内の主要開発言語の割合でJavaの25.4%に次いで2位(16.8%)

IPA「ソフトウェア開発データ白書2010-2011」より

将来も利用し続ける  
宿命にある

【開発リソース】 ワールドワイドでは十分

【COBOL製品】 グローバルスタンダードなCOBOLコンパイラやOSSも存在

**COBOLは現役のプログラム言語である！**



# ユーザー部門の抱える問題

景気後退による  
IT経費削減

レガシー技術者の  
高齢化

レガシーシステムの  
高額な維持費

レガシー製品のEOSL  
(End Of System Life)



投資対効果の低い  
システム更改

ベンダプロプライエタリ  
アーキテクチャ

高いシステム  
更改リスク

高額なシステム  
更改コスト

ベンダロックオン



究極の個別最適  
(製品カスタマイズ)

設計書が無い・  
整合が取れない

ベンダーへの  
責任転嫁

システムに深い  
愛着がある





# 脱レガシーの課題

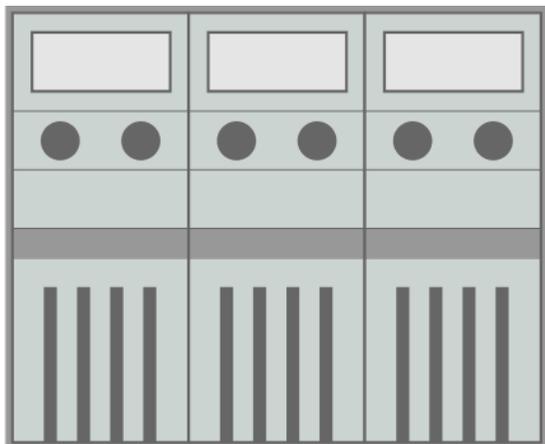


**多額の投資はできない**



**コスト・リスクが増大**

**レガシーシステム**



**従来のモダナイゼーション**



**再構築**



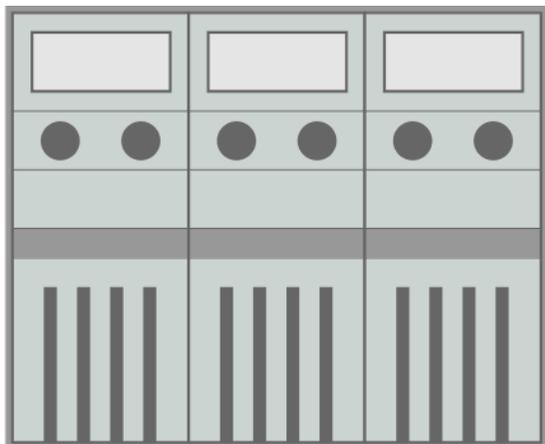
**AsIsで再構築**

**付加価値の追加**

設計書とソースの  
整合が無い

ソースの複雑化

レガシーシステム



再構築

レガシー製品  
提供機能の開発

階層型DB

JCL

特殊ファイルアクセス

Etc.....



どうせ更改するなら.....

あれもやりたい  
これもやりたい....

RDB

OSS

JAVA

web

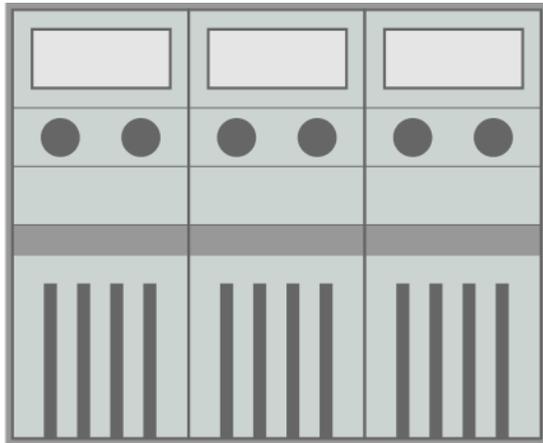
SOA



夢は広がる  
コスト・リスクも広がる....



## レガシーシステム



COBOL以外に  
多種・多様な言語  
(手続き型言語)が存在  
業務・サービス毎に一連の  
処理で記述する言語体系

変換するなら？

## オープン系システム



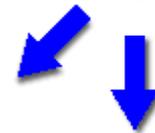
Javaをはじめとした  
オブジェクト指向言語

親和性の高いCOBOLが  
現実的

### レガシー言語を取り込んだCOBOLユーザの拡大

**第一段階：AsIsのリホストによるITコスト削減**

**削減コストでリホスト投資回収**



**第二段階：システム再構築(ToBe)に向けたIT投資**

**ノンコアなシステムの場合、第一段階だけで継続利用も**



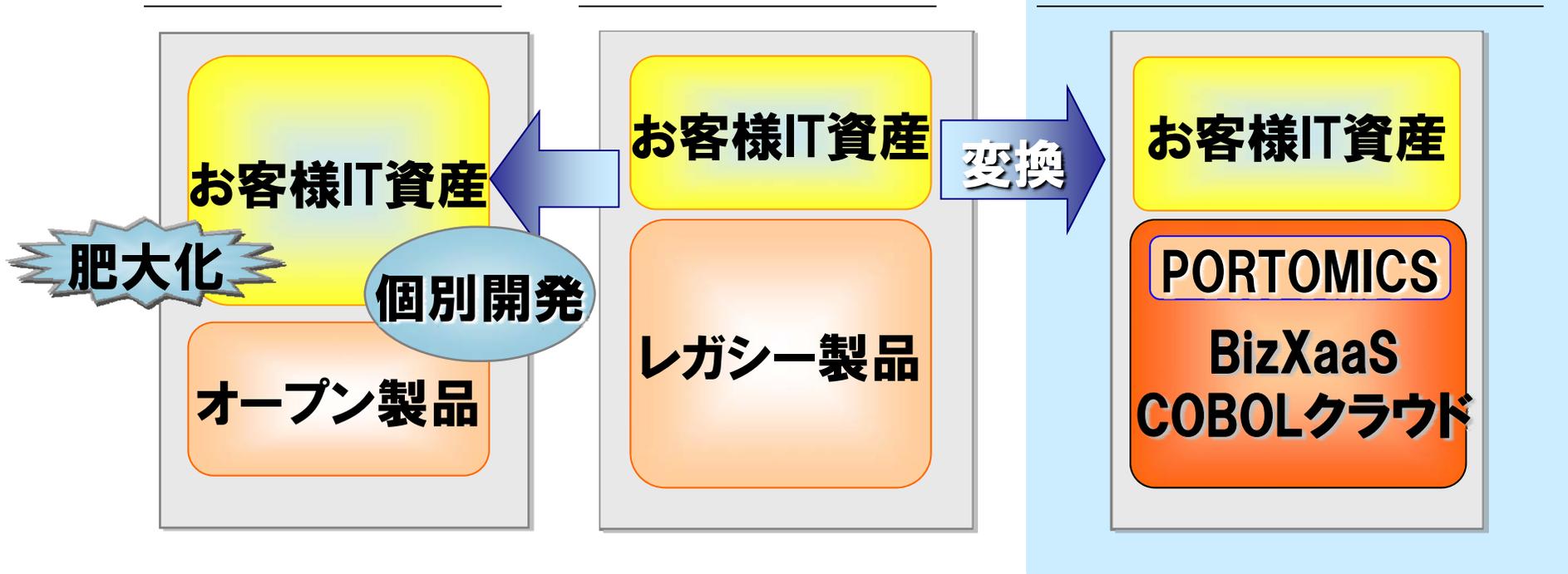


# COBOLクラウドサービスとは？

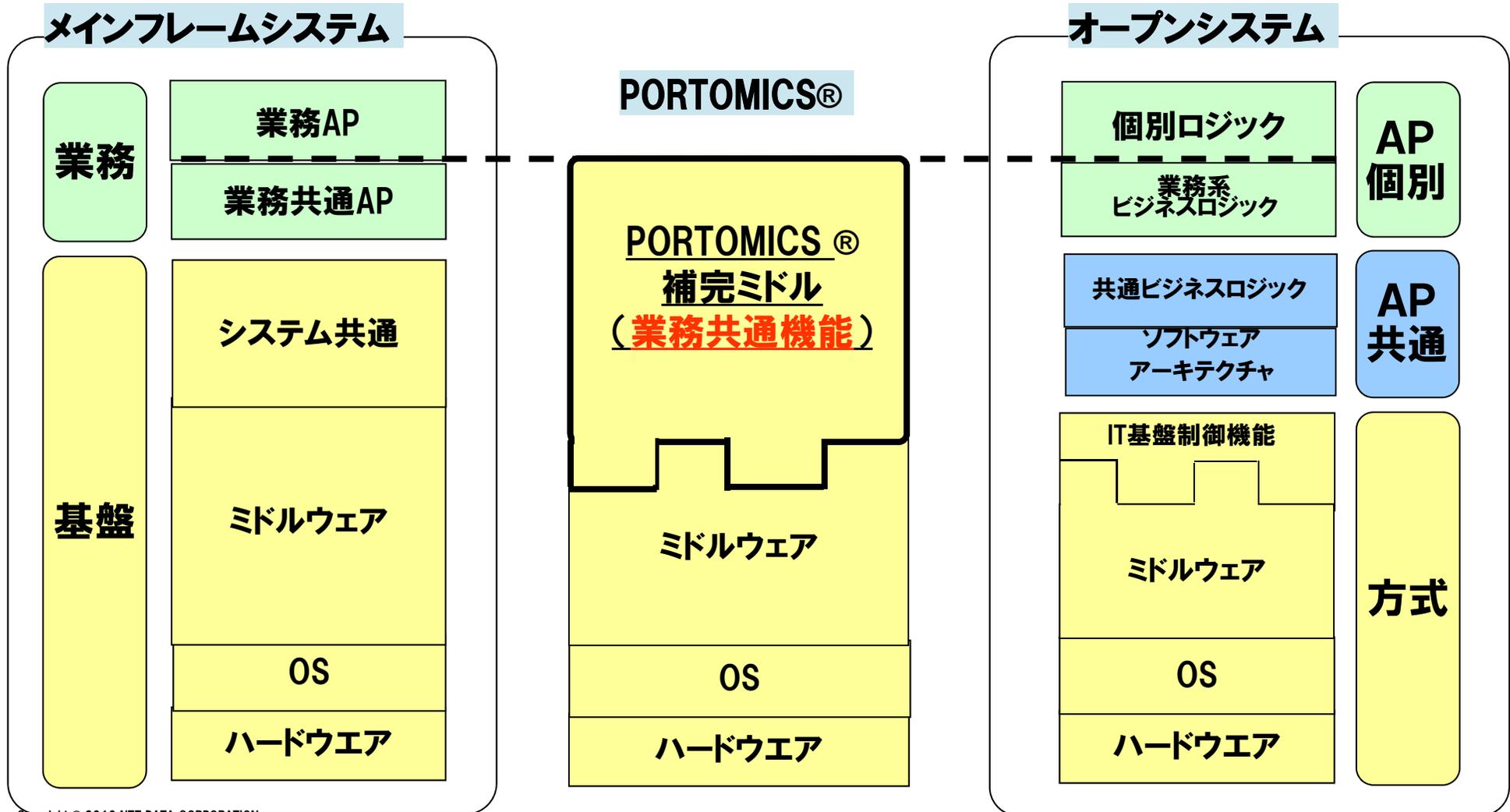
従来のダウン  
サイジング

移行前システム  
(レガシーシステム)

BizXaaS COBOLクラウド  
を用いたダウンサイジング



# ソフトウェア機能構成イメージ



 **ITの所有から利用へ**

 **オンデマンド**

 **エラスティック**

## 「早い」の理由

- ・システムのクラウド提供により  
**システム構築に関する期間が短縮できます**
- ・グローバルなマイグレーションリソースの活用により  
**プログラム変換作業期間が短縮できます**
- ・弊社内外のレガシー専門技術者との連携により  
**迅速なコンサルテーション対応が提案できます**

## 「安い」の理由

- ・レガシー固有機能のPaaS提供により  
**業務プログラム以外の個別機能開発コストが削減できます**
- ・システムのクラウド提供により  
**お客様IT資産(固定資産)の削減が可能です**
- ・専門技術者による現行資産の徹底的な分析と見える化により  
**適正な見積り根拠で提案します**



# 本日のまとめ

- **レガシーシステムユーザは景気悪化、EOSL等の問題で将来ビジョンの策定が急務ですが、決め手がありませんでした**
- **モダナイゼーションの問題は、投資対効果が不明確かつ更改リスクが高く、実施判断が難しい点です**
- **BizXaaSマイグレーションサービスではレガシーアプリケーションの実行環境をクラウド提供することで、更改コスト・リスク共に最適化し、お客様に安心してモダナイゼーションの可否をご判断いただけます**



- **作り直しに夢を描くのは止めよう**  
既存仕様の散逸(仕様が分からない中での作り直して....)  
ゼロからのシステム開発(これまでの歴史の繰り返し)
- **COBOLはプラットフォーム、パラダイムを超える唯一の言語**  
安心して使い続けよう
- **レガシー言語モダナイズで To COBOL需要はますます増大する**

ITに対する従来の価値観を根底から覆す事件が近年立て続けに起きている。

- ・リーマンショックによる世界経済の悪化 → TCO削減
- ・東日本大震災 → BCPの見直し
- ・原発問題 → 電力総量規制対応 → 節電・停電対策・サーバ移転

従来の価値観  
冗長化による拡張性、信頼性

これからの価値観  
省電力、可搬性

- 新たなIT評価基準
- ・電力消費量
  - ・移設のし易さ
  - ・立上げ/下げ時間
    - ・耐震構造
    - ・設置条件

IT部門の方々は非常なご苦勞を  
されていると思います。  
少しでもお役に立てることができればと  
思います。

変える力を、ともに生み出す。

NTT DATAグループ



**COBOLクラウドでレガシーシステムユーザ様のお役に立ちたい**

NTTデータの総合クラウドサービス

**BizXaaS**

※BizXaaSは、株式会社NTTデータの登録商標です。

## 定義

**「システムの存在が国民の社会生活や企業活動にとって必要不可欠なもの」**

システムが停止した場合、国民の社会生活や企業活動に致命的な影響が生じるITサービス  
たとえば、金融機関(銀行、証券、決済)、社会インフラ(旅客運輸、電力、通信)、官庁系  
(年金、国家予算、国防)

## 構成要素

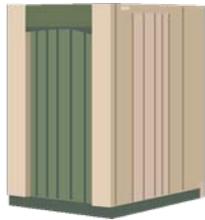
- ・24時間365日連続稼働していること **【高可用性】**
  - ・障害等によるサービス停止、中断の防止 **【高信頼】**
    - ・**り障範囲の局所化、拡大防止**
    - ・**原因・対策の明確性と即時性**
    - ・**処理結果のACID(原子性、一貫性、独立性、永続性)保証**
  - ・高い処理能力、高負荷時の安定性 **【高性能】**
    - ・重要処理の優先実行
    - ・想定外の外的要因(突発負荷、異常操作等)に対する安定性
- 等々、システム毎に要件レベルは異なる

平時の要件充足性以外に  
異常(想定外)時の業務処理の  
安定性、一貫性、即時性が求  
められる

これまではメインフレームの利用が大半

メインフレーム、オープンシステムはそれぞれ異なるパラダイムで発展してきたアーキテクチャである。利用者がアーキテクチャをチョイスする際にはASISの実現方式に捕らわれずにパラダイムの違いを考慮した要件整理が必要である。

## メインフレームシステムとは



単一ベンダ製品  
(ハード・ソフト)  
によるシステム

・単一ベンダの統一された製品構成のため統一的なインターフェースによるシステム構築・運用管理が可能

・特定ベンダに依存する  
・初期費用が高い

集中型システム

## オープンシステムとは



複数ベンダ製品(ハード・ソフト)の組合せが可能

・要件に適した製品を選択できる  
(新機能や新規格等に容易に対応可能)  
・初期費用が安い

・異なる製品を組み合わせるため、統一感あるシステムを実現するためには、十分な検証と工夫が必要

分散型システム

利点

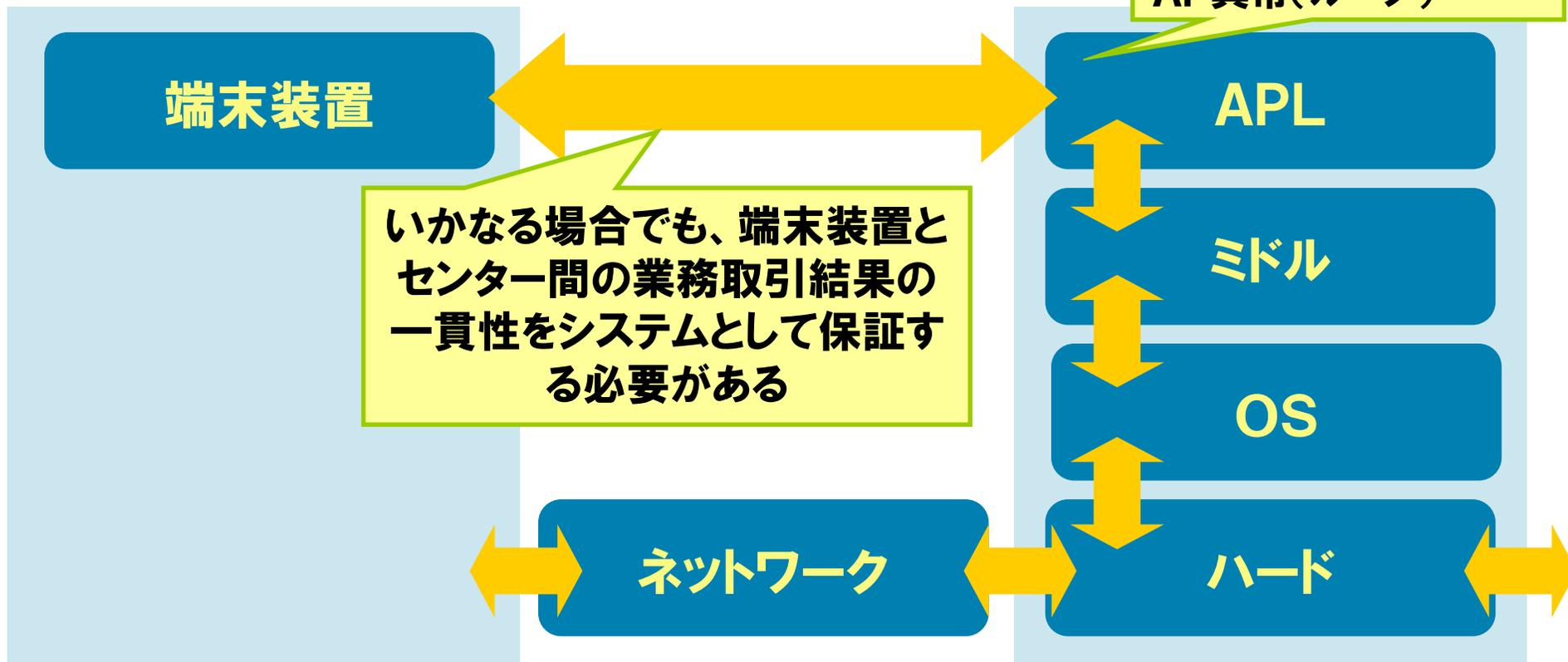
欠点

得意分野

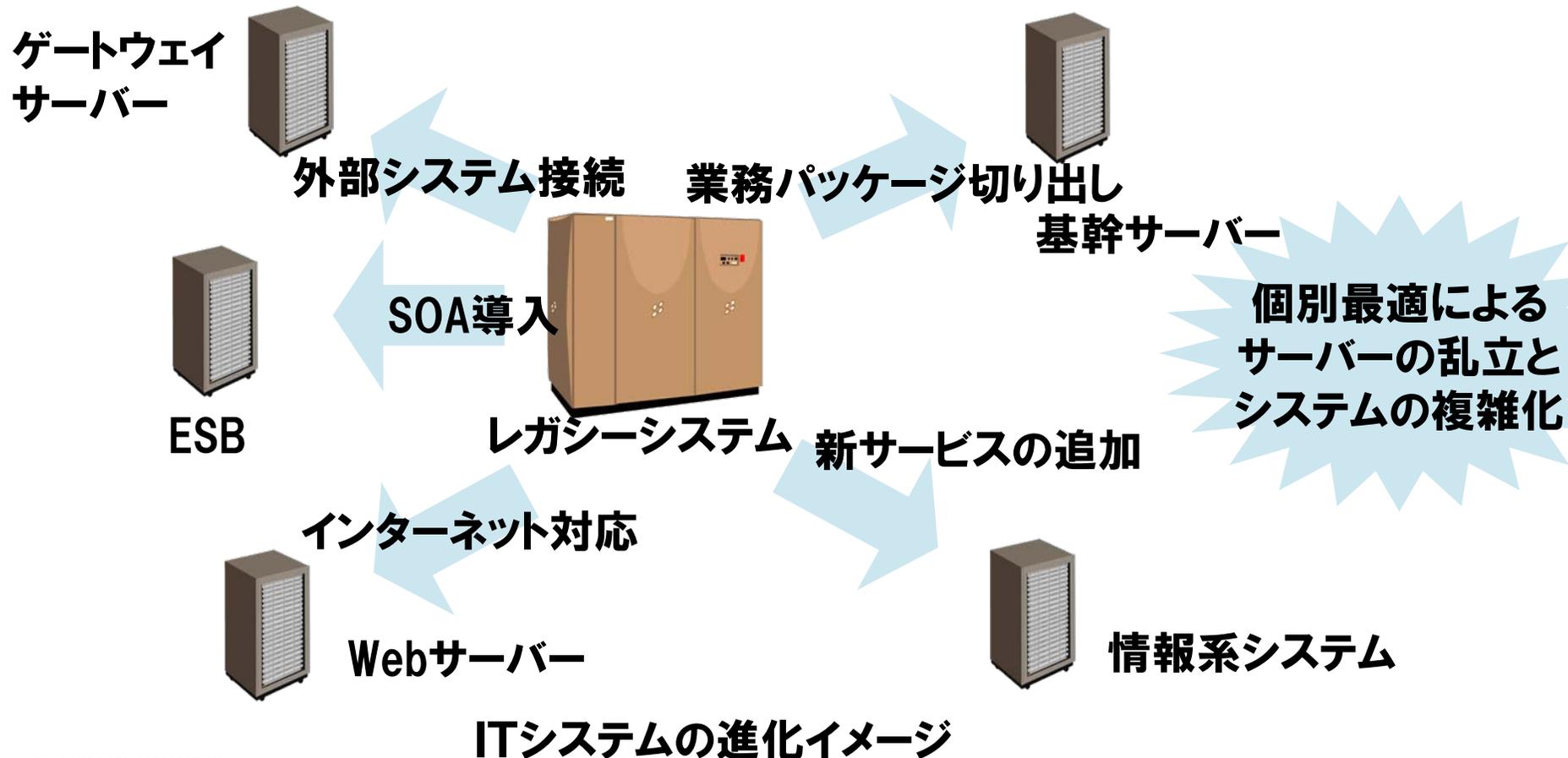
オープン系システムでMissionCriticalシステムを実現するためには、モジュラー型アーキテクチャで不得意な異常時の業務処理の安定性、一貫性を解決する必要がある。

そのためには、個々のレイヤ機能の強化ではなく、業務処理を含めたトータルの連携強化が必要。

- ・ハード/ネットワーク故障
- ・オーバーロード(過負荷)
- ・DBデッドロック
- ・AP異常(ループ)



基幹系システムは当初はメインフレーム上にオールインワンで構築後、時代のニーズ対応するため個別最適で拡張を行った結果、レガシーシステムを中核に複雑に発散を続けている。



PORTOMICS®では、前述のMissionCriticalシステムに求められる異常時の業務処理の一貫性を保証するための各種異常処理呼び出し出口を設けることで、異常発生時のきめ細かい対応を実現しています。

