

HITACHI
Inspire the Next

情報システムに拡張性・生産性向上を もたらずSOA実践アプローチ

株式会社 日立製作所 ソフトウェア事業部 先端情報システム研究開発部
田坂 光伸

Contents

1. SOA実践における課題
2. 日立が考える解決へのアプローチ
3. 段階的なシステム最適化
4. 統合システム構築基盤Cosminexusのご紹介

uVALUE

1

SOA実践における課題

1-1

企業を取り巻く環境の変化

企業を取り巻く環境は、日々変化し多様化している。

変化への適応力を高め、経営スピードを高める企業戦略が重要

Web2.0型ビジネス

規制緩和・自由化

総務省 u-Japan政策

業界再編



企業の社会的責任(CSR)

日本版SOX法(J-SOX法)

ITに求められている役割も変化してきている。

経営の効率化に加え、高度化も求められている。

経営の高度化

- いかに変化対応力を進化させるか。

事業運営の最適化

- いかに成果を生み出すか。

業務プロセスの効率化

- いかにBPRを実践するか。

業務支援

- いかに関ツールを使いこなすか。

《効率化》

- ・部分最適
- ・現場対応
- ・コスト重視
- ・業務プロセスのIT化
- ・問題解決



《高度化》

- ・全体最適
- ・環境変化対応
- ・効果重視
- ・業務プロセスの改善
- ・戦略展開

情報システムの課題： IT部門に求められている「業務プロセス改革」

ビジネスユーザーがIT部門に求めている企画提案

- 「ITを活用した業務プロセス改革」が圧倒的

基幹業務システム再構築に直面している企業は8割を超す。

- 再構築の理由1位は「業務の効率化、プロセス改革のため」

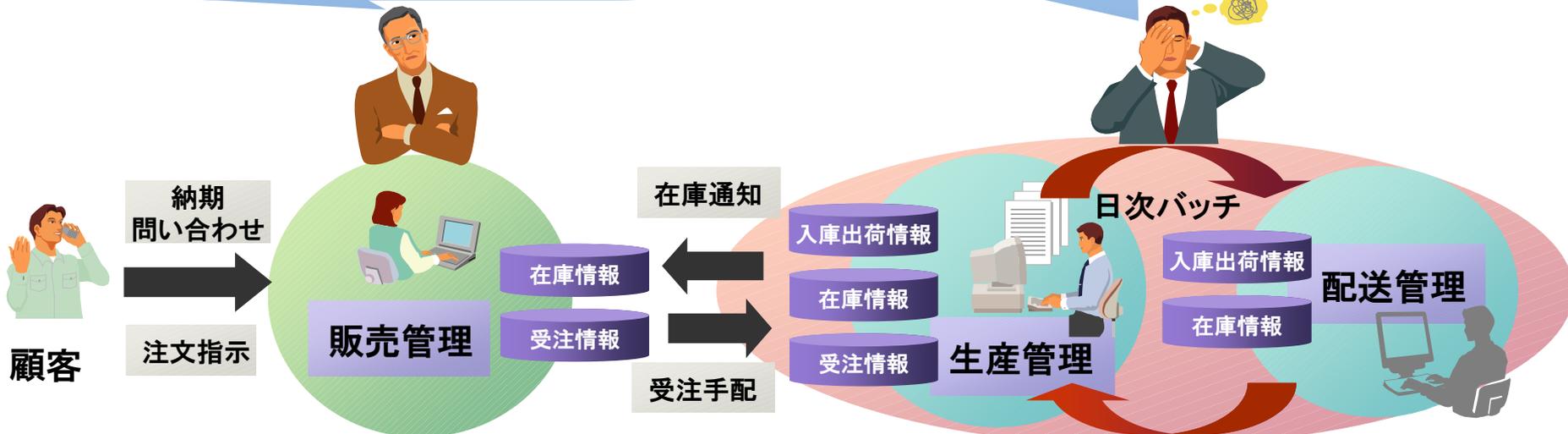
JUAS(日本情報システム・ユーザ協会) ユーザ企業IT動向調査 2006より

ビジネスユーザーの不満

「リードタイムが長い」
「納期回答に時間がかかる」
「システムが使いにくい」

IT部門の悩み

「改修に手間がかかる」
「システムをつなげるのも大変」
「運用も複雑」

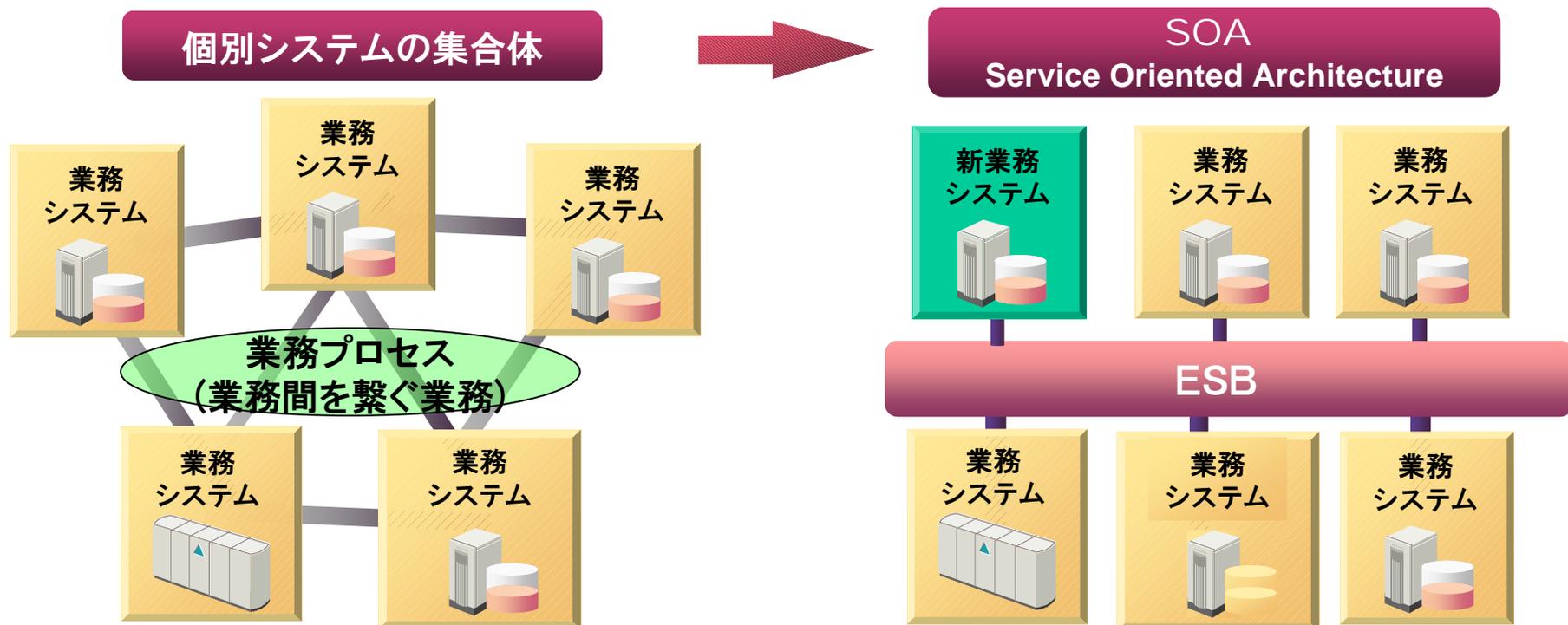


ビジネスニーズに合った業務プロセスの最適化や変化即応の実現

個別業務のIT化は行われているが、業務間を繋ぐ業務(業務プロセス)における

- ビジネスニーズに合った最適化
- ビジネス変化に即応したシステム改修が課題になっている。

- 既存業務システムのサービス化と、それらを組合わせた
- 業務プロセスの適切なシステム化を実現し、業務プロセスにおいて、ビジネスニーズに合った最適化と変化即応を実現

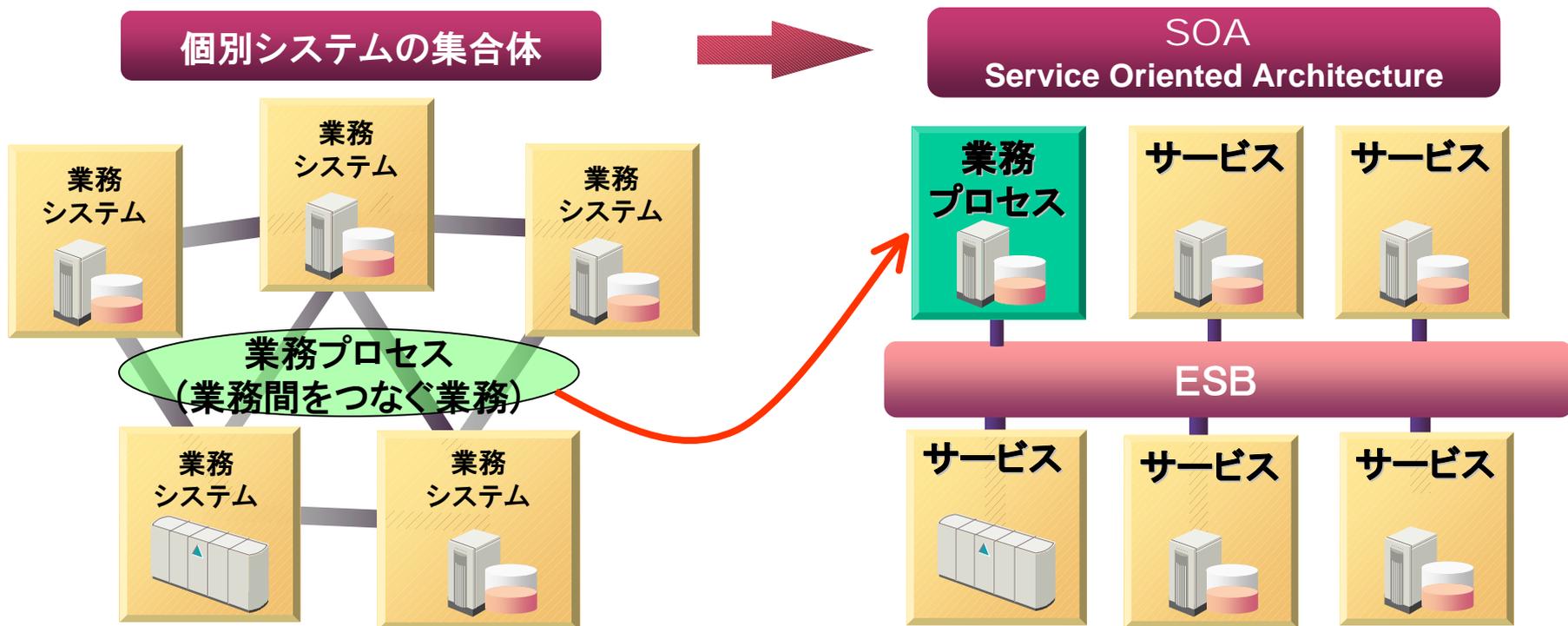


ビジネスニーズに合った業務プロセスの最適化や変化即応の実現

個別業務のIT化は行われているが、業務間をつなぐ業務(業務プロセス)における

- ビジネスニーズに合った最適化
- ビジネス変化に即応したシステム改修が課題になっている。

- 既存業務システムのサービス化と、それらを組合わせた
- 業務プロセスの適切なシステム化を実現し、業務プロセスにおいて、ビジネスニーズに合った最適化と変化即応を実現

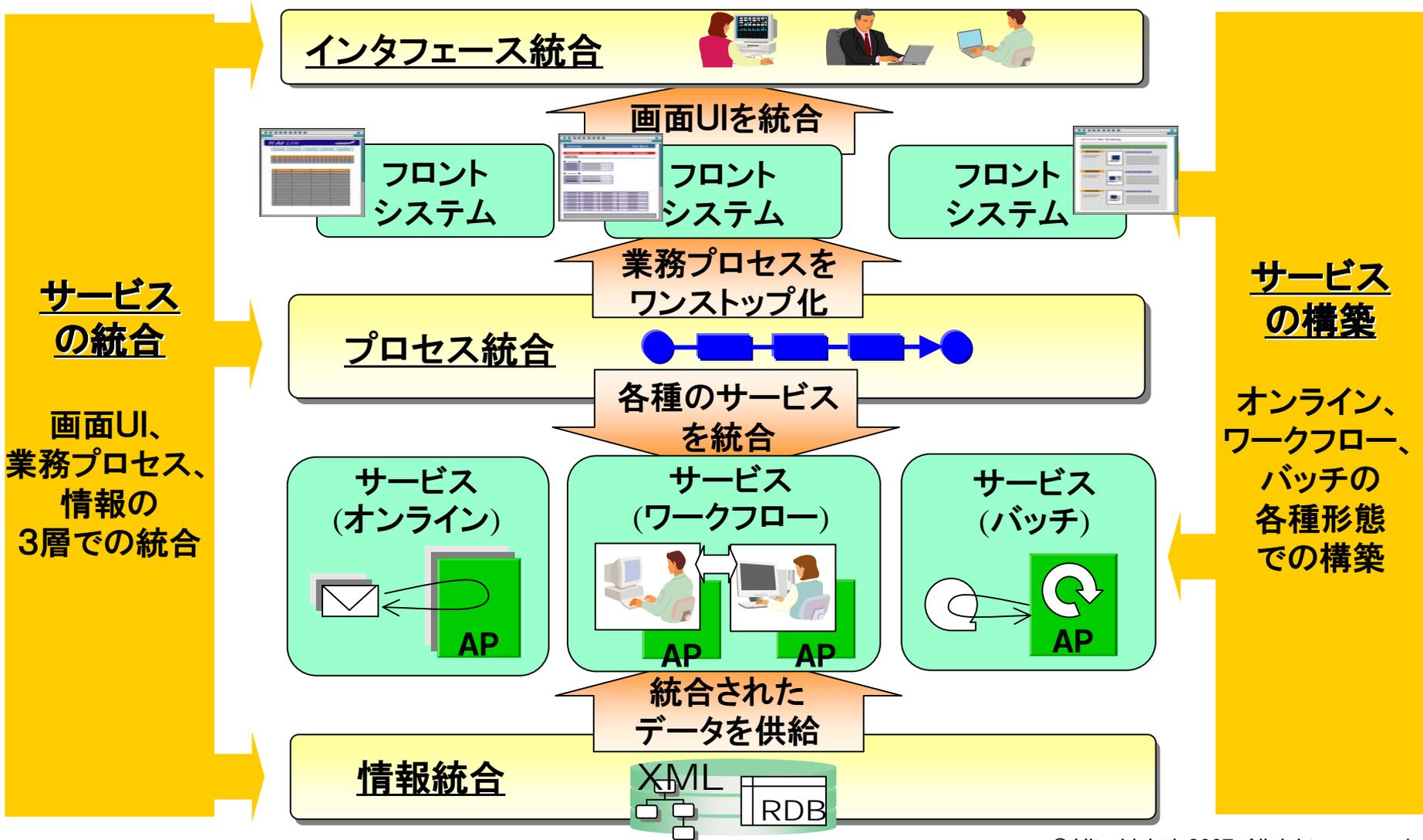


- (1)多様なサービス統合・サービス構築形態への適用
- (2)段階的なシステム最適化

SOA実践における課題と解決指針

(1)多様なサービス統合・構築形態への適用

多様なサービス統合・サービス構築の全体像

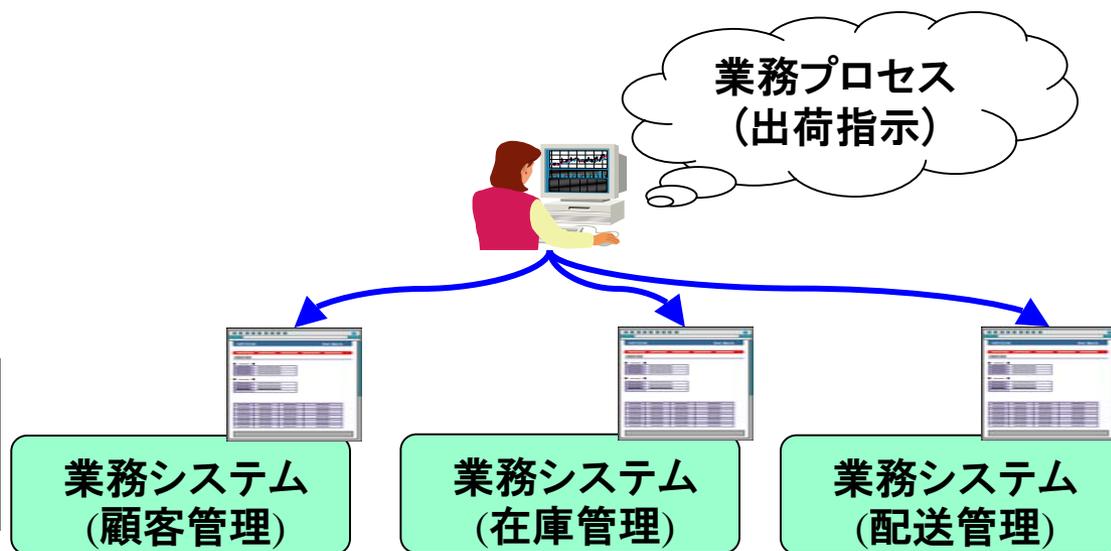


SOA実践における課題と解決指針： (1)多様なサービス統合・構築形態への適用

業務プロセスの最適化・変化即応のためには、
業務プロセスに沿った統合に加えて、インタフェースや情報の統合の併用が不可欠

課題：

業務間をつなぐ業務プロセスが非効率。
複数システムを個別に使用したり、
人的判断不要なのに人手で連携

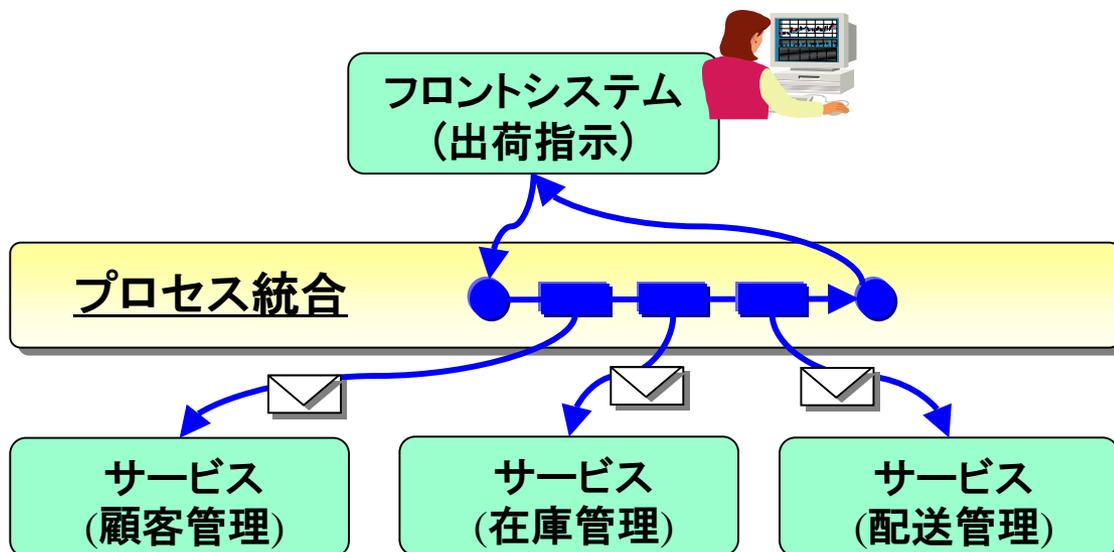


SOA実践における課題と解決指針： (1)多様なサービス統合・構築形態への適用

業務プロセスの最適化・変化即応のためには、
業務プロセスに沿った統合に加えて、インタフェースや情報の統合の併用が不可欠

解決指針：

既存システムを組み合わせ、
業務プロセスのワンストップ化・
自動化を実現する。



SOA実践における課題と解決指針： (1)多様なサービス統合・構築形態への適用

業務プロセスの最適化・変化即応のためには、
業務プロセスに沿った統合に加えて、インタフェースや情報の統合の併用が不可欠

課題：

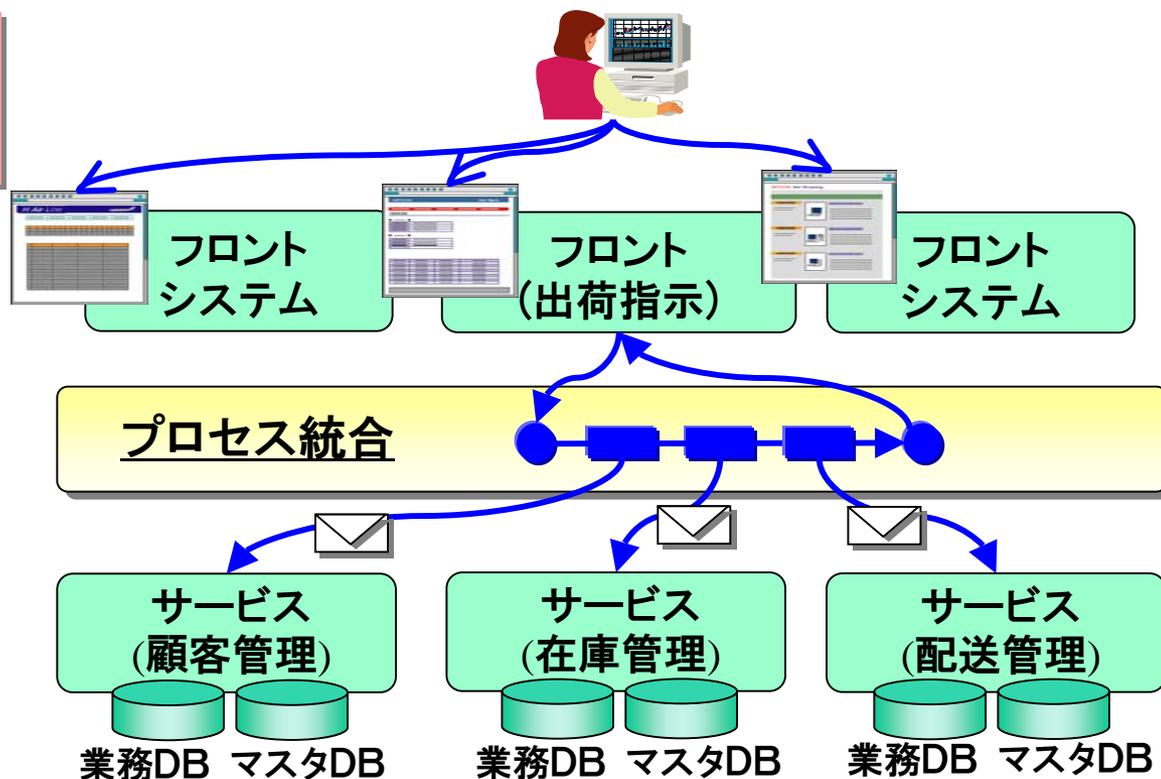
一作業者は、複数業務を担当。
複数システムのUIを別々に起動し、
非効率

解決指針：

既存システムを組み合わせ、
業務プロセスのワンストップ化・
自動化を実現する。

課題：

業務プロセスを通じて参照される
マスタデータの整合性確保が課題



SOA実践における課題と解決指針： (1)多様なサービス統合・構築形態への適用

業務プロセスの最適化・変化即応のためには、
業務プロセスに沿った統合に加えて、インタフェースや情報の統合の併用が不可欠

解決指針：

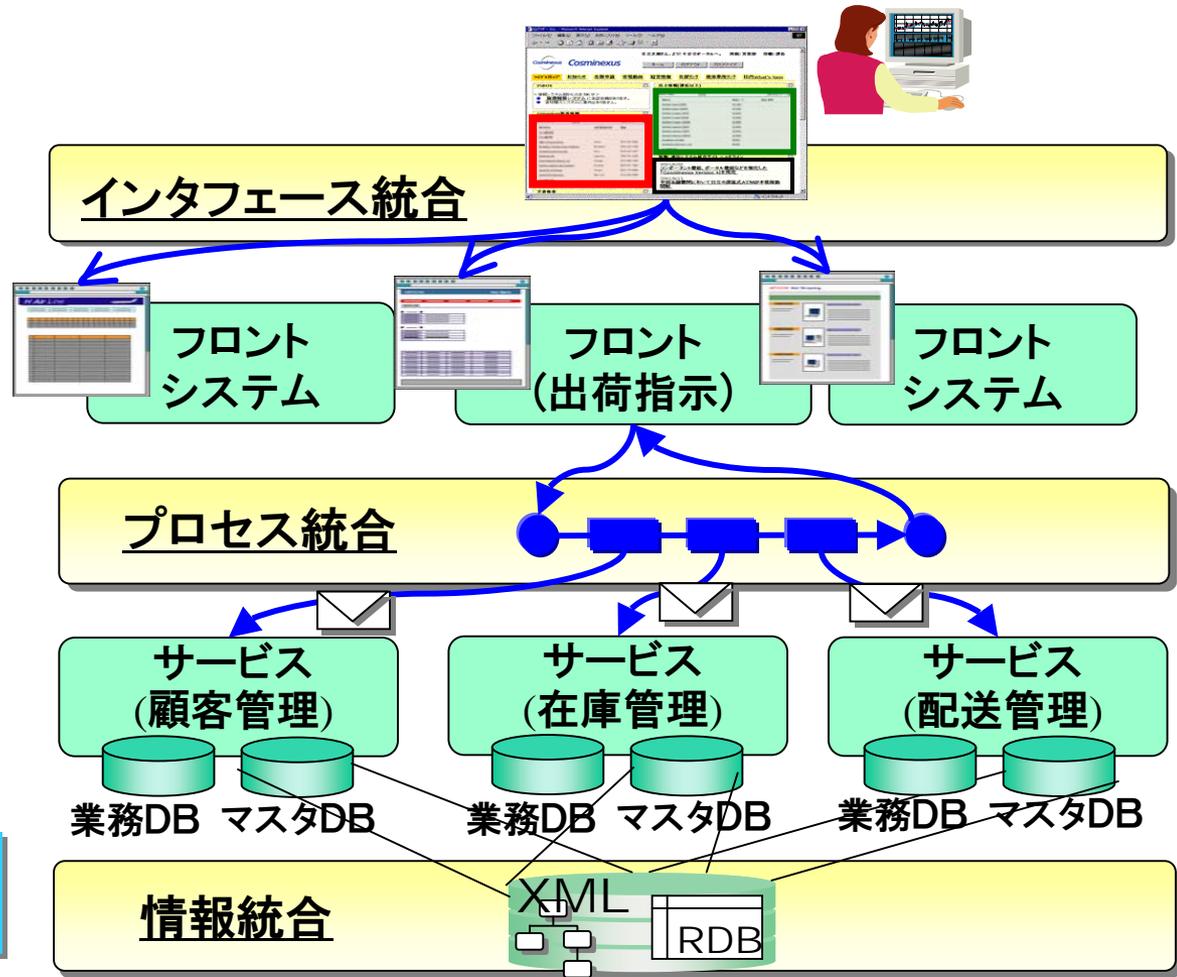
作業員の担当業務に合わせて、
複数システムのUIを、
一画面に統合する。

解決指針：

既存システムを組み合わせ、
業務プロセスのワンストップ化・
自動化を実現する。

解決指針：

マスタを統合し、既存システムへの
データ供給を実現。



SOA実践における課題と解決指針： (1)多様なサービス統合・構築形態への適用

業務間をつなぐ業務(業務プロセス)における

- ビジネスニーズに合った最適化
- ビジネス変化に即応したシステム改修が課題

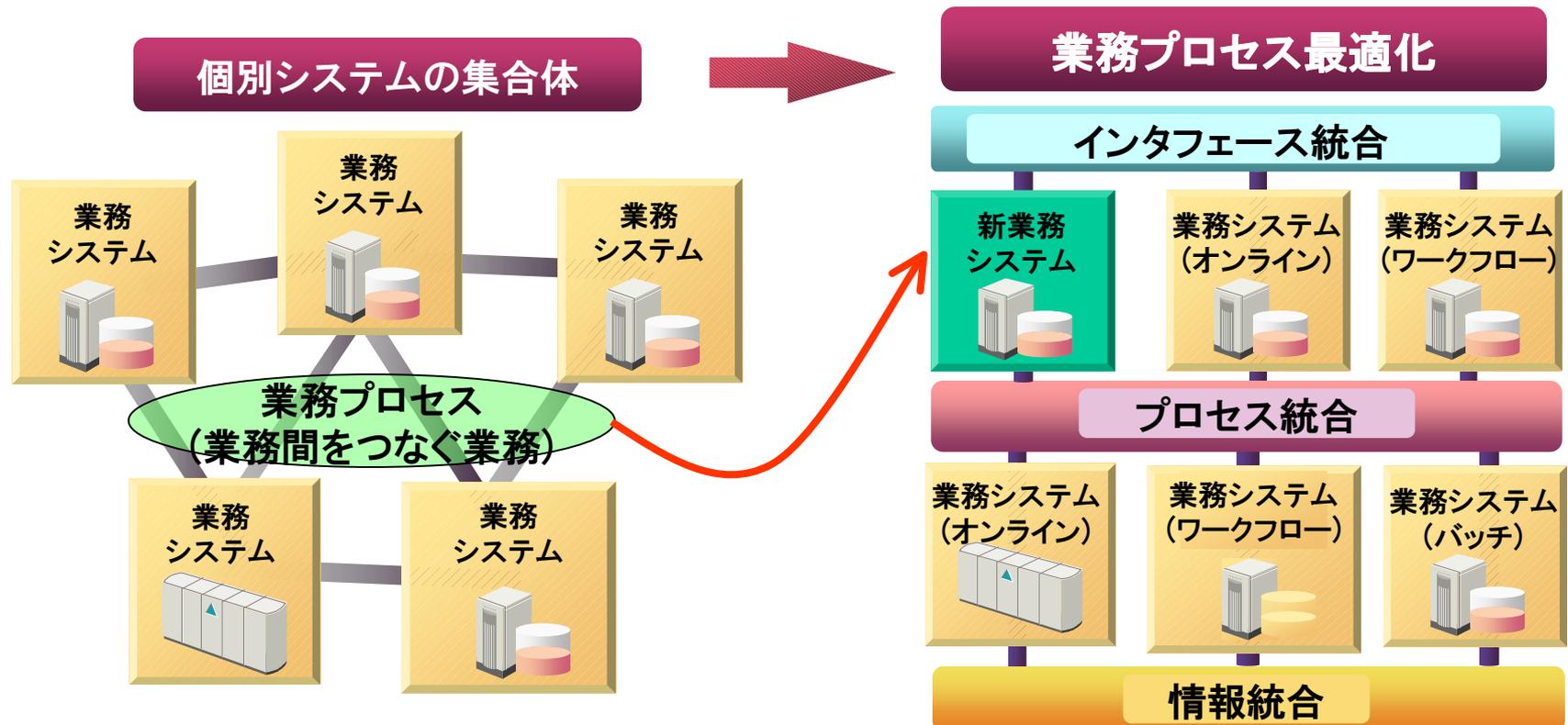
業務プロセスのシステム化における
ビジネスニーズに合った最適化と変化即応の実現

【実践の指針】

多様な形態の業務システムを多様な形態で統合

【更なる課題】

サービス統合の生産性と拡張性の確保が不可欠

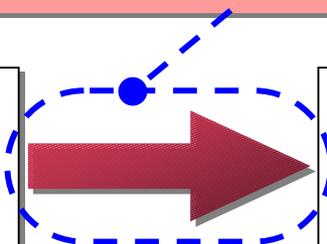
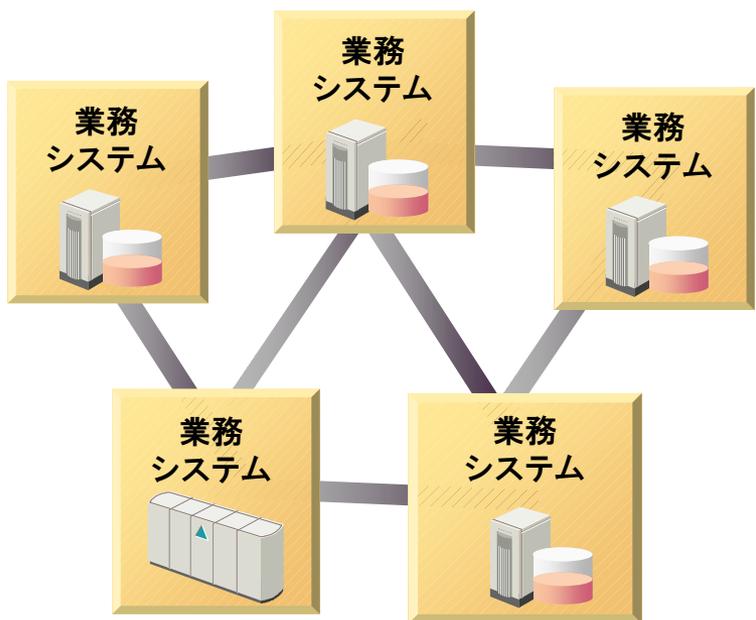


SOA実践における課題と解決指針： (2)段階的なシステム最適化

課題：

全体最適なシステムへの全面更改 ⇒
設計難易度が高く、リスク(費用・業務移行等)も大きい。
開発期間が長期化。効果が出るまでに時間がかかる。

AsIs: 個別システムの集合体



ToBe: 業務プロセス最適化

インターフェース統合

新業務
システム

業務システム
(オンライン)

業務システム
(ワークフロー)

プロセス統合

業務システム
(オンライン)

業務システム
(ワークフロー)

業務システム
(バッチ)

情報統合

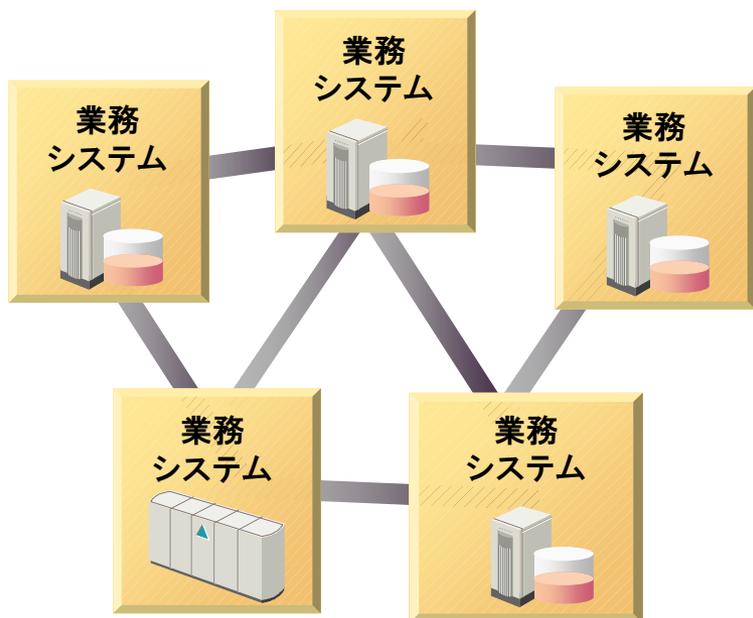
SOA実践における課題と解決指針： (2)段階的なシステム最適化

解決指針：

全体最適化へ向けて段階的にシステム拡張を進められる現実的アプローチ

- ・現行システムを活用しながら、業務プロセスのシステム化、新規システムの追加、現行システムの変更・刷新などを段階的に実施
- ・段階的な移行で、効果を早期に創出し、かつ、コスト・時間を最適化(導入リスクを低減)
- ・ゴールとなる情報システムアーキテクチャも、段階的に見直して最適化していく。

AsIs: 個別システムの集合体



フェーズ 1

フェーズ 2

フェーズ 3

ToBe: 業務プロセス最適化

インタフェース統合

新業務システム

業務システム
(オンライン)

業務システム
(ワークフロー)

プロセス統合

業務システム
(オンライン)

業務システム
(ワークフロー)

業務システム
(バッチ)

情報統合

SOA実践における課題と解決指針： (2)段階的なシステム最適化

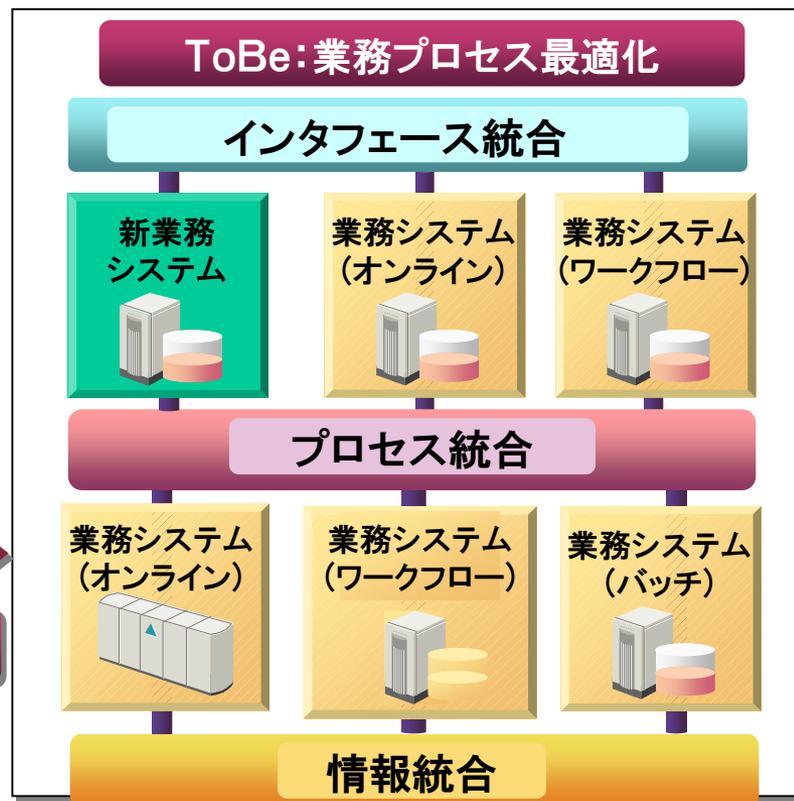
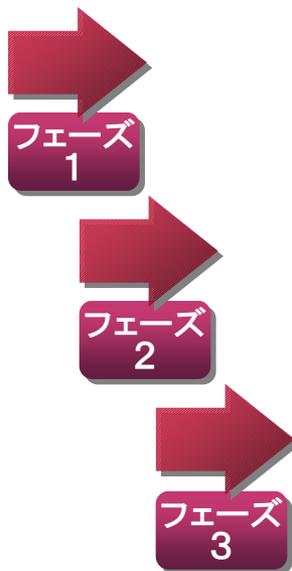
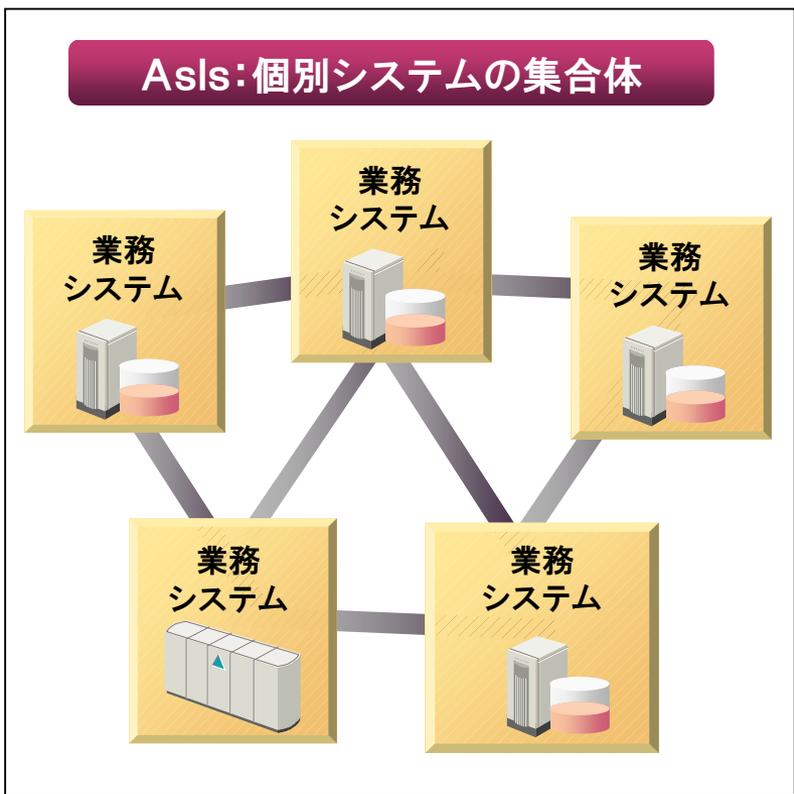
業務プロセスのシステム化におけるビジネスニーズに合った最適化と変化即応の実現

【実践の指針】

全面刷新ではなく、段階的なシステム最適化により、効果早期創出とリスク回避

【更なる課題】

段階的に業務システムを追加・改変できる**拡張性**が不可欠



uVALUE

2

日立が考える解決へのアプローチ

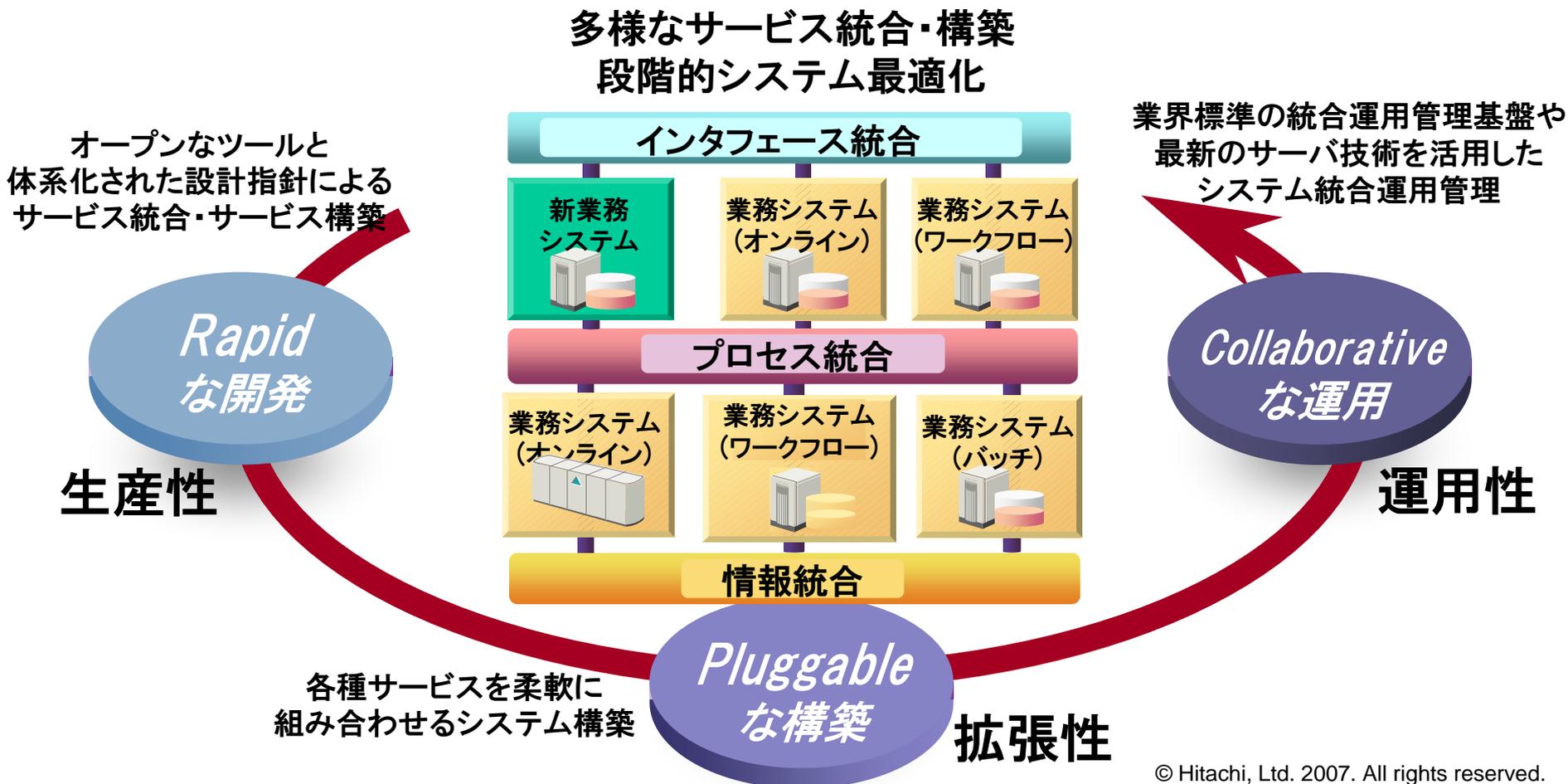
HITACHI
Inspire the Next

業務プロセス最適化の実践におけるIT課題に対して、ソリューションを提供します。

- ✓ 多様な形態の業務システムを多様な形態で統合
- ✓ 段階的なシステム最適化



- 統合・構築の生産性と拡張性
- 統合したシステムの運用性確保



業務プロセス最適化の実践におけるIT課題に対して、ソリューションを提供します。

- ✓ 多様な形態の業務システムを多様な形態で統合
- ✓ 段階的なシステム最適化



- 統合・構築の生産性と拡張性
- 統合したシステムの運用性確保

統合システム構築基盤 *Cosminexus*

基盤技術 + SIノウハウ

サービス統合
サービス構築
基盤製品

リファレンス
アーキテクチャ
適用ガイド

基盤製品だけでなく、
システム設計ガイドラインの提供による
生産性・拡張性・運用性の向上

Rapid
な開発

生産性

Collaborative
な運用

運用性

Pluggable
な構築

拡張性

uVALUE

2.1

Rapid — 生産性 —

多様な処理形態に適切な基盤機能と共に、SIノウハウを体系化した設計ガイドラインを提供し、システム構築の開発生産性を向上

要件定義、業務設計のガイドラインを提供

ビジネス
環境の
変化

要件
定義

業務設計
アプリケーション設計

システム設計

システム構築

サー
ビス
統合

インターフェース統合

プロセス統合

情報統合

サー
ビス
構築

オンライン

ワークフロー

バッチ

システム
運用

システム設計のガイドラインを提供

統合・構築の各種処理形態に適切な
開発環境・実行環境を提供

2.1-2

リファレンスアーキテクチャと適用ガイドの提供

システムパターンに基づきシステム設計・システム構築のガイドラインを提供

(1)システム設計のリファレンスとして、サービス統合・構築のシステムパターンを定義

(2)システムパターンを適用した方式設計やサイジング・環境構築等の指針を提供

要件定義・業務設計

システム設計

システム構築

システム開発手法
(日立およびパートナー様)

要件定義・業務設計ガイド

- ビジネスプロセス設計
- アプリケーション設計

Cosminexus

1

SOAリファレンス
アーキテクチャ

2

SOA適用ガイド

- システム方式設計
- サイジング・環境構築

基盤機能

インタフェース統合

プロセス統合

情報統合

サービス構築

2.1-3

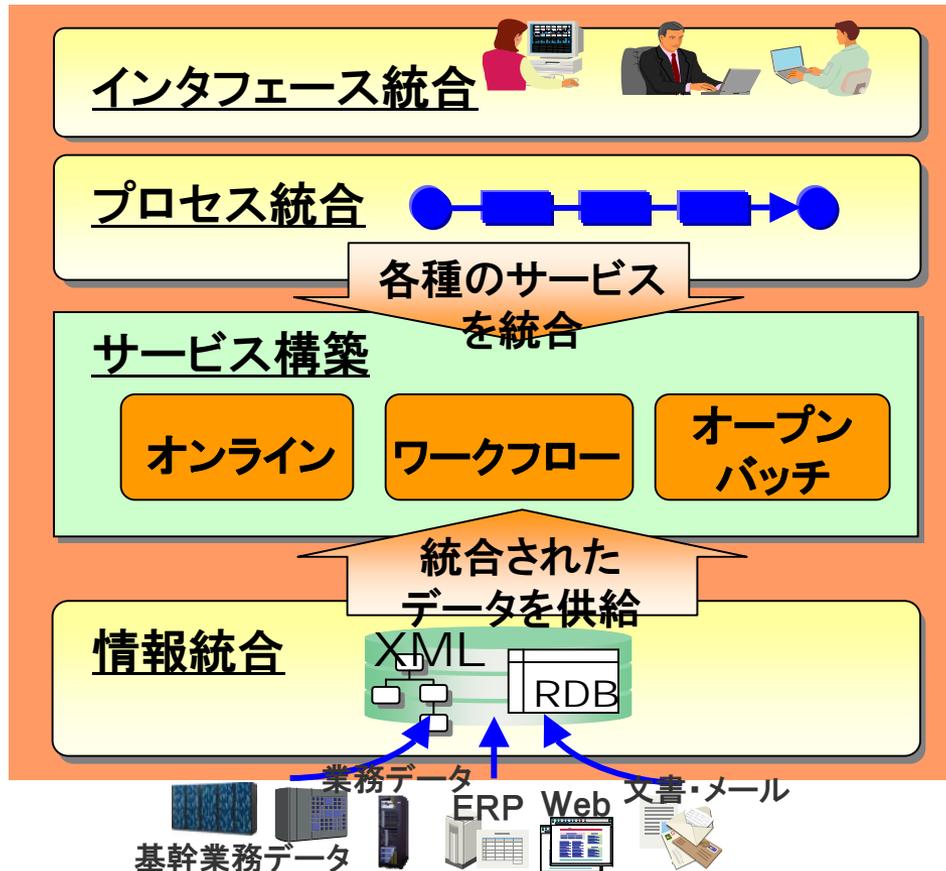
SOAリファレンスアーキテクチャ

システム構築に必要な各種システムパターンを明確化し、提案・設計を効率化

◆多様なシステム要件に応じて、サービス統合・構築のシステムパターンを定義

- インタフェース・プロセス・情報の観点で、業務間の連携を実現し、拡張性を確保。
- オンラインに加え、ワークフロー、バッチといった多様なサービス構築形態も対応。

概念図



システムパターン

インタフェース統合編

プロセス統合編

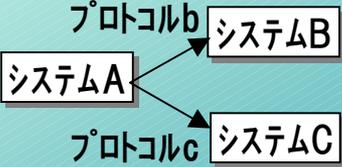
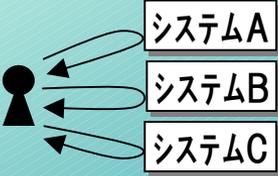
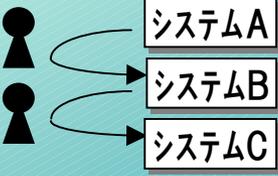
情報統合編

サービス構築編

2.1-4

システムパターン一覧（抜粋：プロセス統合の例）

課題を解決し、やりたいことを実現するシステムパターンを定義

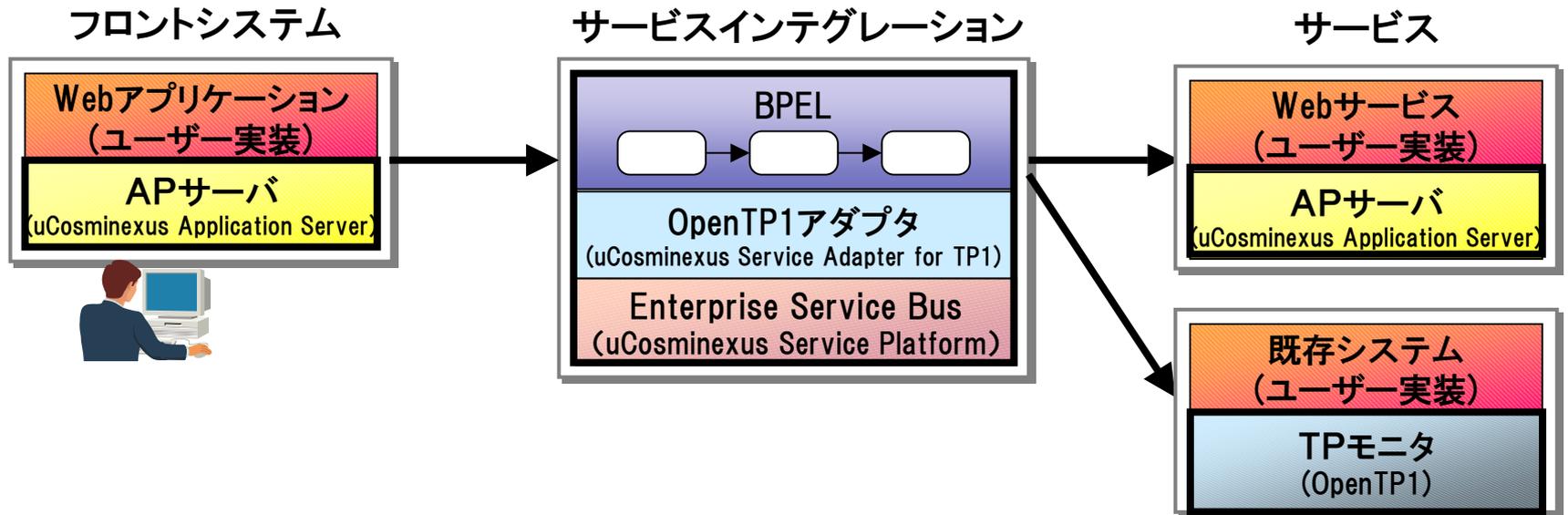
#	課題	目的	実現の考え方	システムパターン
1	<p>既存・新規システムが混在し、開発が複雑</p>  <pre> graph LR A[システムA] -- "プロトコルb" --> B[システムB] A -- "プロトコルc" --> C[システムC] </pre>	システム間連携 開発方法の統一	ESBにより、既存システム接続の差異を吸収し、呼出元には単一の開発方法(Webサービス)を提供する。また既存システム刷新しても、呼出元への影響を極小化する。	ESBパターン
2	<p>一人の作業者が複数のシステムを使い分けており、非効率</p>  <pre> graph LR P((人)) --> A[システムA] P --> B[システムB] P --> C[システムC] </pre>	複数システムを 操作する作業者の 効率化・自動化	BPELにより、システム間連携を実現し、フロント業務でのシステム操作をワンオペレーション化する。	ワンストップサービスパターン
3	<p>システム間連携に作業者の介入が大きく、非効率</p>  <pre> graph LR P1((人)) --> A[システムA] P1 --> B[システムB] P2((人)) --> B[システムB] P2 --> C[システムC] </pre>	システム間連携 の効率化・自動化	ビジネスプロセスに沿ったシステム間連携を、BPELにより自動化する。JP1/AJS2との連携により、ファイル到着などのイベントを契機にBPELを暗黙実行。	イベント駆動パターン

2.1-5

システムパターン例：ワンストップサービスパターン

課題を解決し、やりたいことを実現するシステムパターンを定義

プロセス統合編		ワンストップサービスパターン		
システム概況	課題	やりたいこと	実現の考え方	
部門業務はシステム化されているが、フロント業務の作業者は、複数の部門業務システムを手で使い分けている。	複数システムを個別に使い分けるため、処理に時間がかかる。また手作業により、ミスが多発している。	フロント業務におけるシステム操作を省力化して業務を効率化したい。	プロセス統合基盤を導入し、BPELによりシステム間連携を自動化し、フロント業務でのシステム操作をワンストップ化	



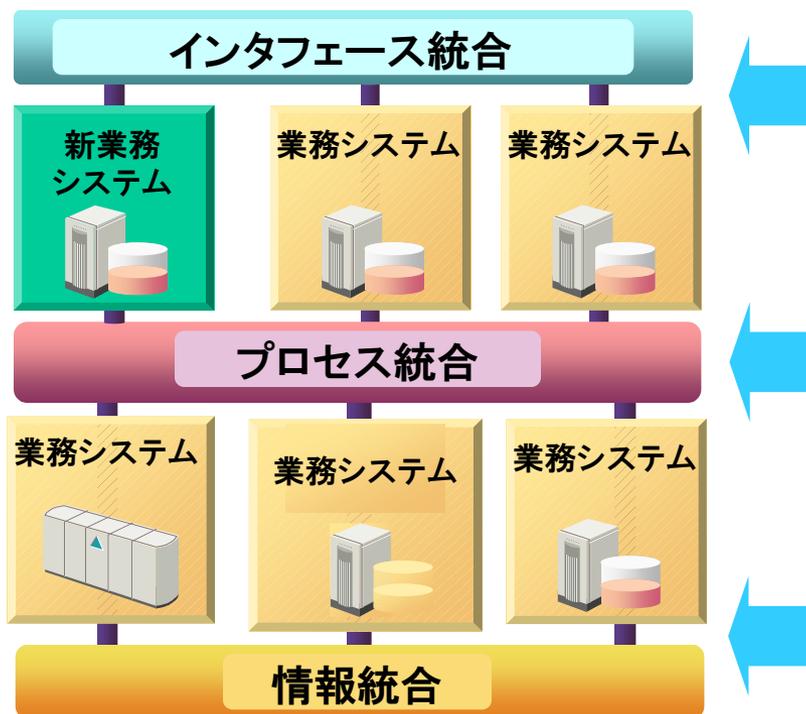
uVALUE

2.2

Pluggable — 拡張性 —

拡張性の高いサービス統合技術とシステム設計ガイドラインにより、 多様な形態での統合と段階的システム最適化を実現

- リファレンスアーキテクチャを通じて、システムパターンの組み合わせによる段階的拡張
- 業務システムの段階的な追加や刷新が容易（後から追加や変更が簡単にできる）



- 複数画面統合の定義や変更が容易
（レイアウト、統合する画面部分切り出しなど）
- SaaSと企業内システムの画面統合が可能

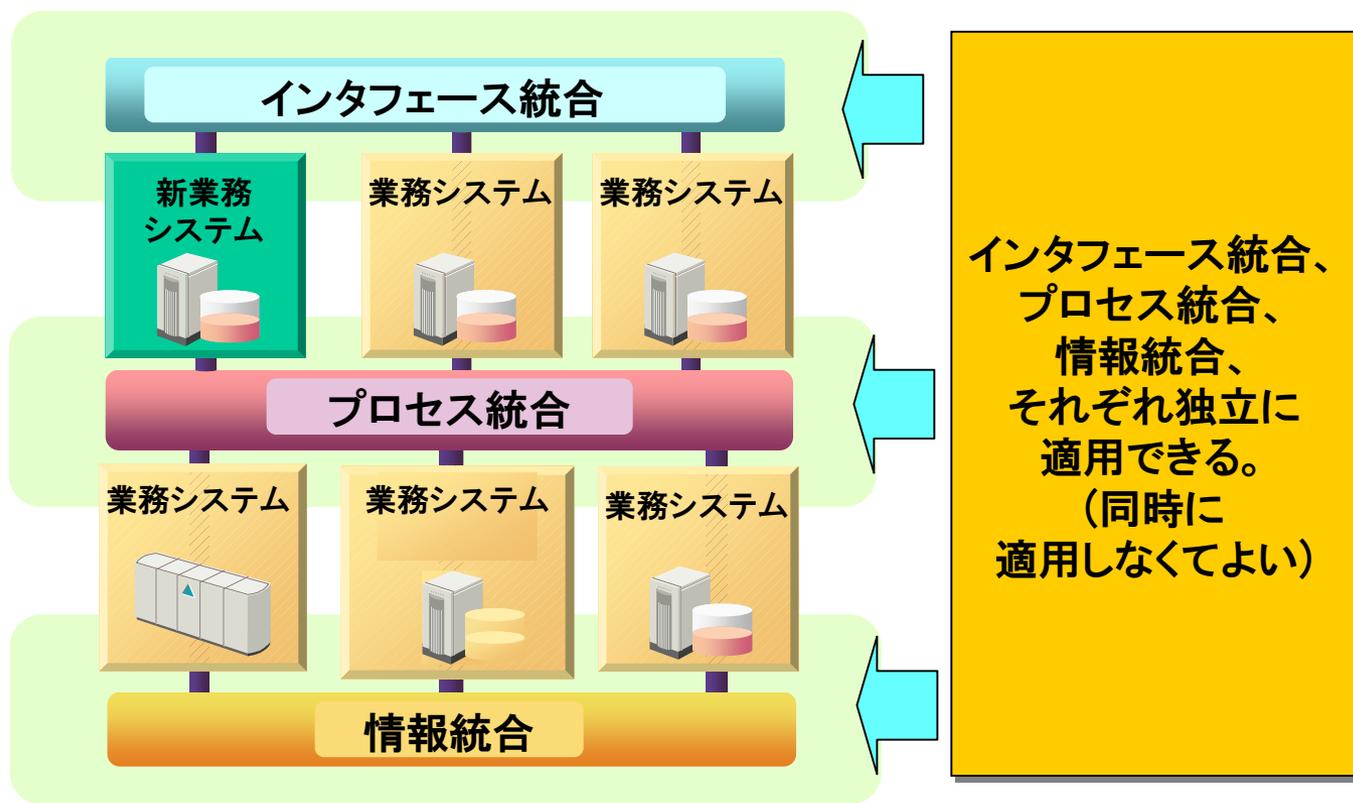
- 業務プロセスの定義や変更が容易
（フロー定義、データ形式の変更など）
- 様々な形態の業務システムの統合が可能

- データ仮想化の定義や変更が容易
（データ集約などの定義・変更）
- 各種のデータソースの統合が可能

段階的システム最適化に適した サービス統合アーキテクチャ (1)

(1) インタフェース統合・プロセス統合・情報統合を部分的・段階的に実施可能

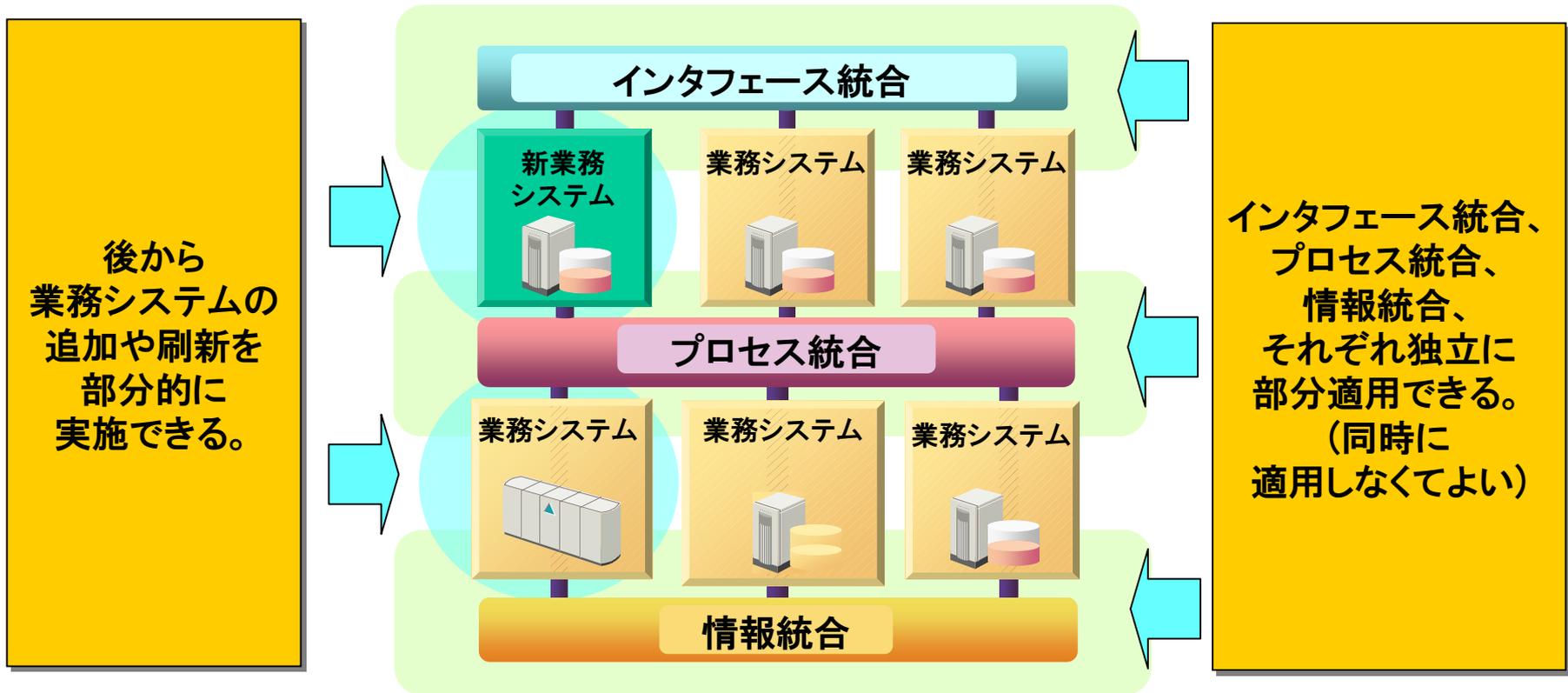
- 統合技術として相互独立性が高く、組合せによる効果も享受できる。



段階的システム最適化に適した サービス統合アーキテクチャ (1)

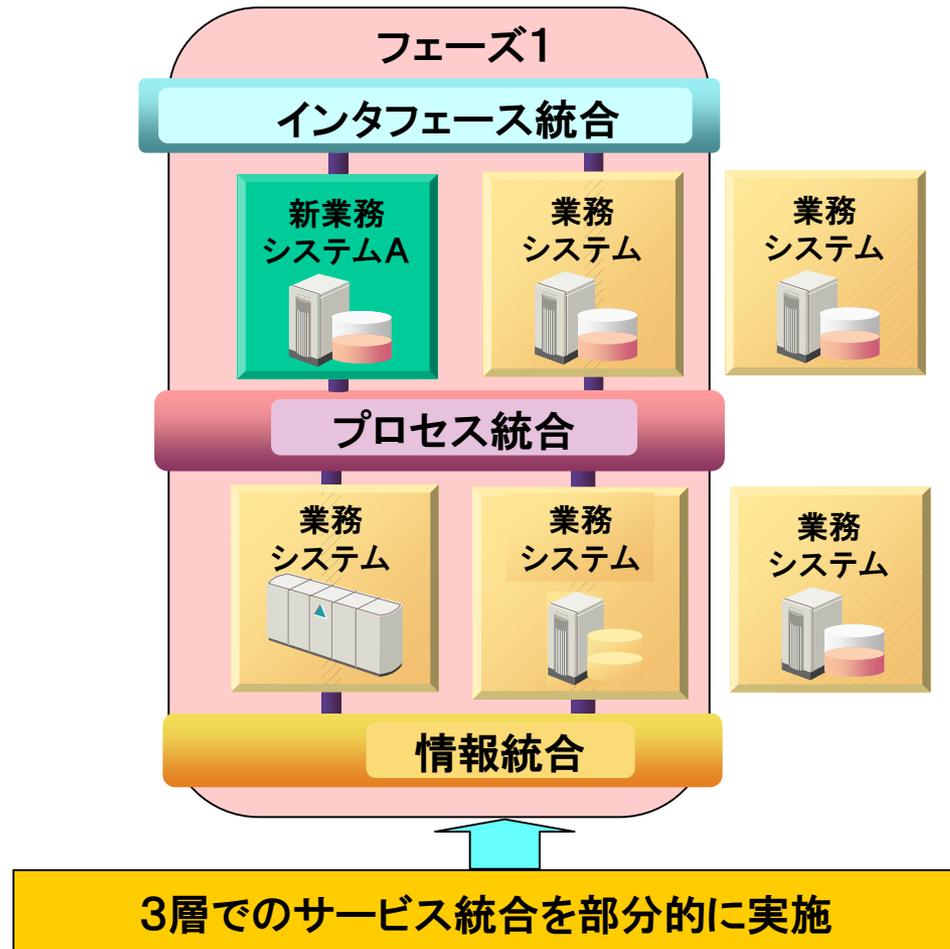
(1) インタフェース統合・プロセス統合・情報統合を部分的・段階的に実施可能

- 統合技術として相互独立性が高く、組合せによる効果も享受できる。
- 各統合層において、業務システムの追加や刷新を部分的に実施可能

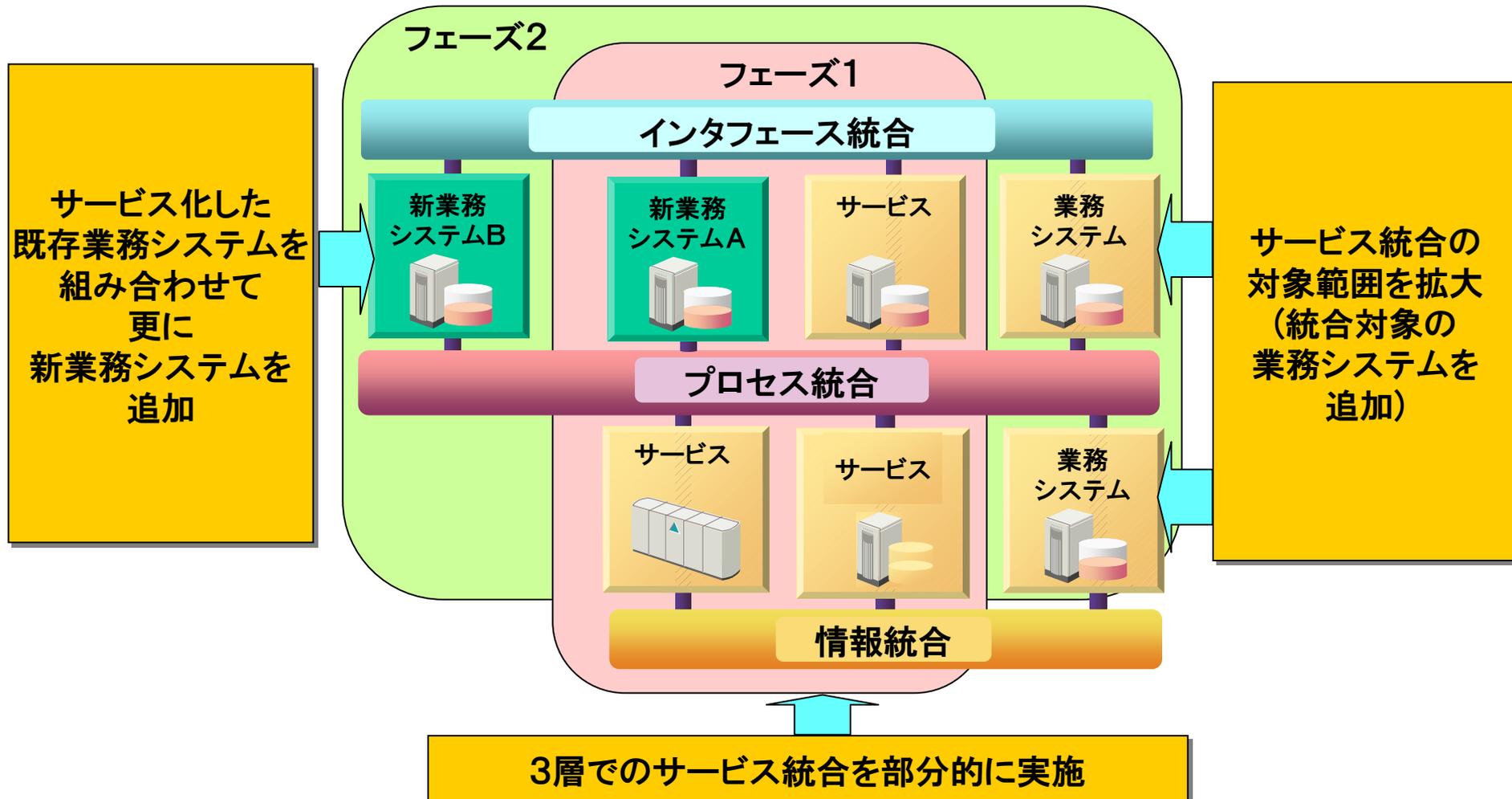


段階的システム最適化に適した サービス統合アーキテクチャ (2)

(2) サービス統合の対象範囲を段階的に拡大していくことが可能

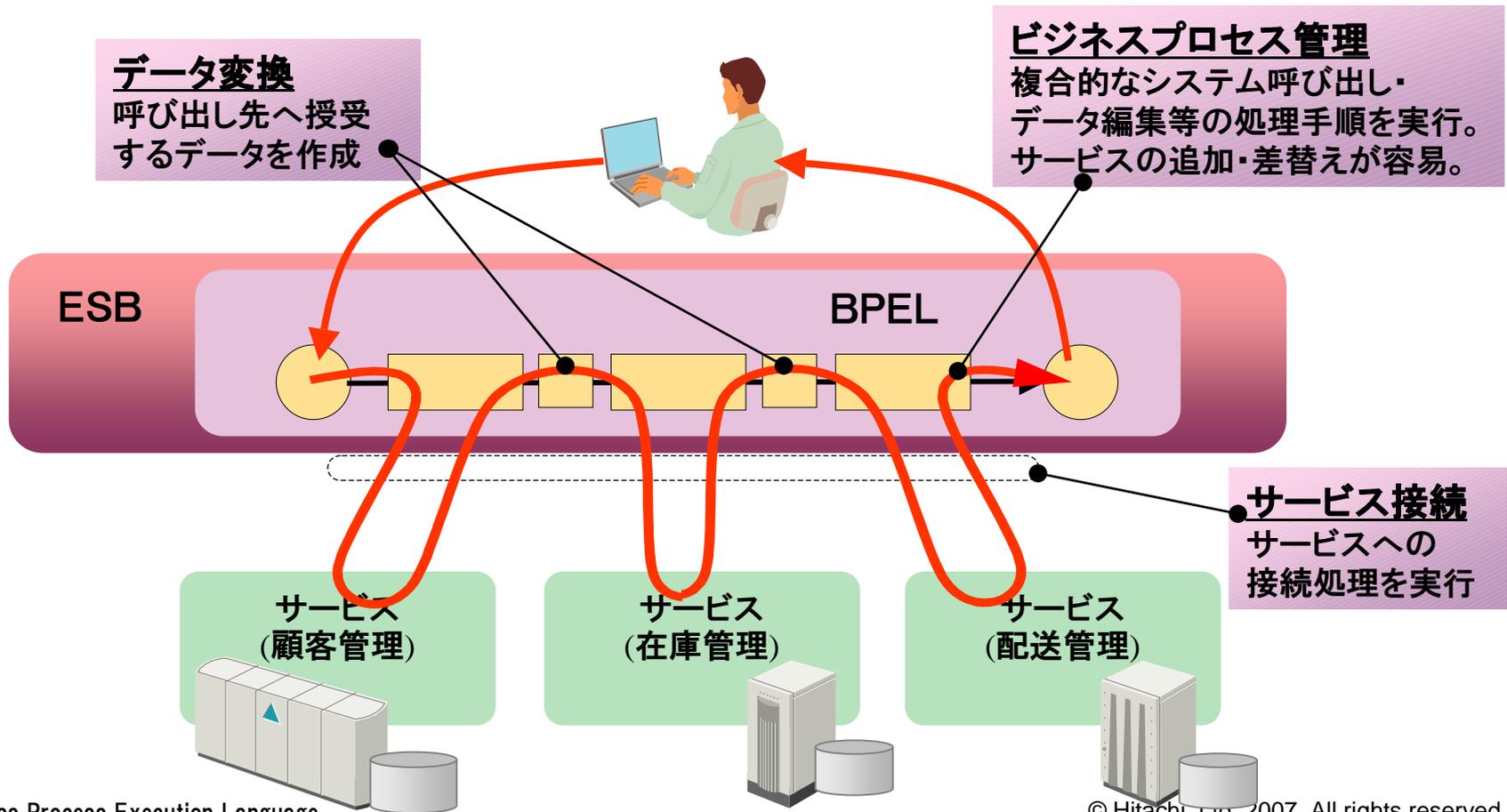


(2) サービス統合の対象範囲を段階的に拡大していくことが可能



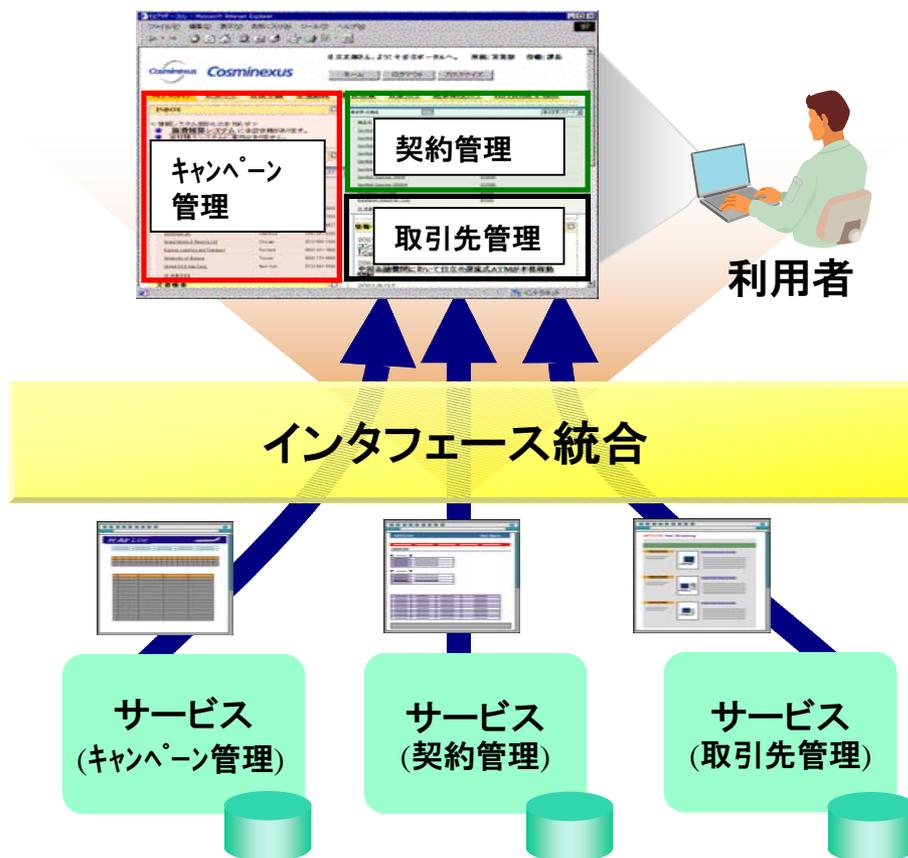
ポイント

- ビジネスプロセス(BPEL)により、複数サービスを組み合わせ、新サービスを容易に構築できる。
- プラガブルなサービス連携を実現する柔軟なESB
既存システムのサービス化、新システムへの差し替えを段階的に実現可能



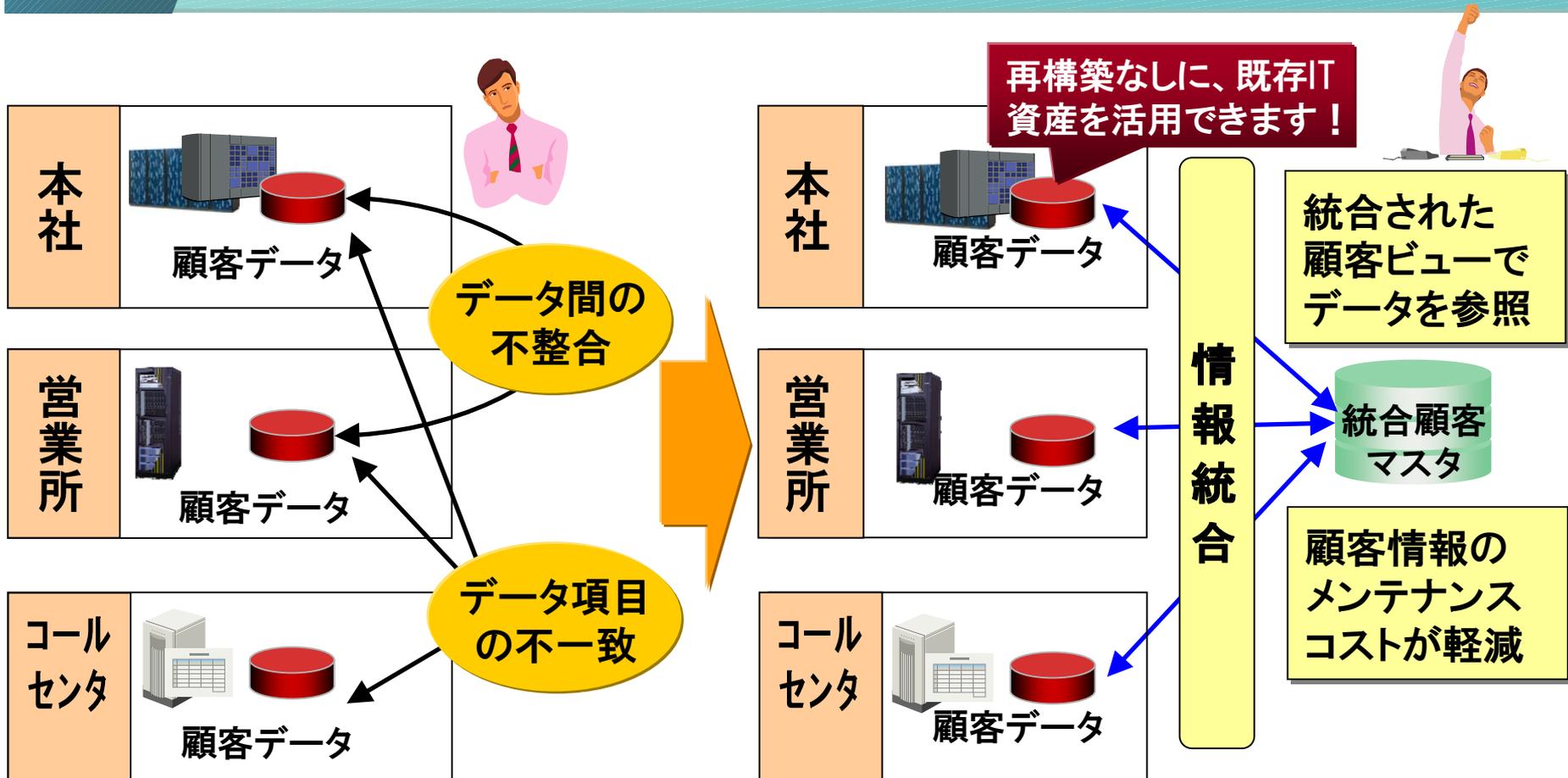
ポイント

- 複数サービスの画面インターフェースを統合し、業務ポータルを実現
利用者の担当業務に応じて、複数業務の画面インターフェースを統合
既存業務システムを活用した統合を実現。段階的な統合も可能
- 画面間での連携・データ引継ぎの簡易開発を実現



ポイント

- 複数システムに散在する顧客情報などマスタデータの一元管理と、既存システムのアプリケーション改修の極小化の双方を実現
- 既存アプリケーションに対し、マスタデータへのアクセス方法を変更しない。既存システムのマスタ実体有無や必要鮮度により、連携方式を選択可能



uVALUE

3

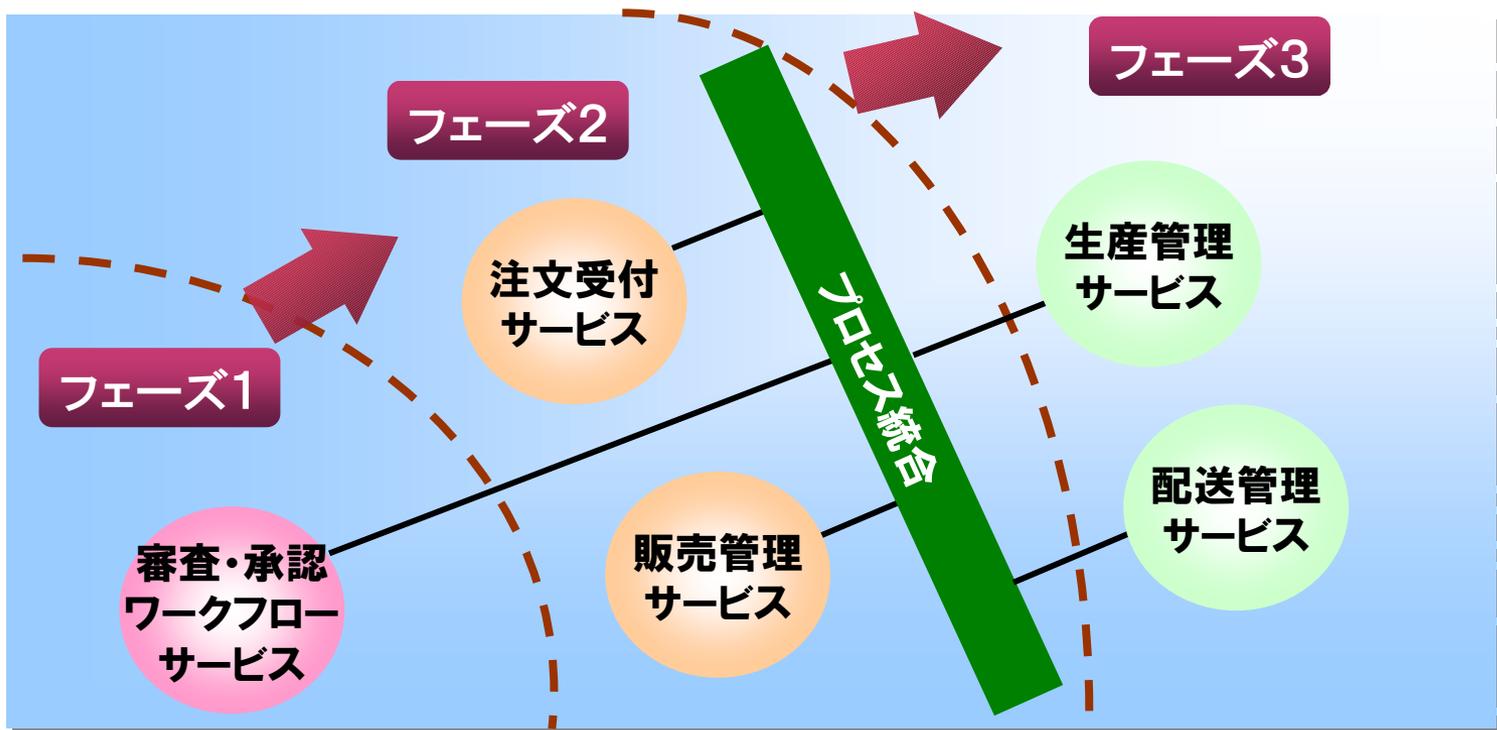
段階的なシステム最適化 —システムパターン紹介と適用例—

3-1

段階的なシステム最適化の考え方

ポイント

- 個別の課題をポイントソリューションで解決することから着手。個別に小さく始めて、早期に課題解決、効果を実感
- 段階的に、全体の業務プロセス最適化に向け拡張を進める
プロセス統合などサービス統合技術の拡張性を活用



段階的システム最適化への適用シナリオ —いくつかのシステムパターンを例にして説明—

システムパターン

フェーズ1

フェーズ2以降

(1) ワンストップサービス
パターン

(2) ビジネスプロセス
自動化パターン

既存システムを組み合わせ
て業務プロセスをシステム化

業務プロセスX

プロセス統合

既存
システムA

既存
システムB

既存
システムC

業務プロセスシステム化範囲拡大や
既存・新規システムの刷新・追加

業務プロセスX 業務プロセスY ……

プロセス統合

既存
システムA

既存
システムB

既存
システムC

新規
システム

フェーズ1

(3) 対話ワークフロー
パターン

個別の業務システム課題の
解決のため、
当システムを刷新

既存
システムA

既存
システムB

既存
システムC

業務プロセスシステム化や
他システムの刷新・追加

uVALUE

3.1

ワンストップサービスパターンの適用

3.1-1

ワンストップサービスパターン: 課題と実現の考え方

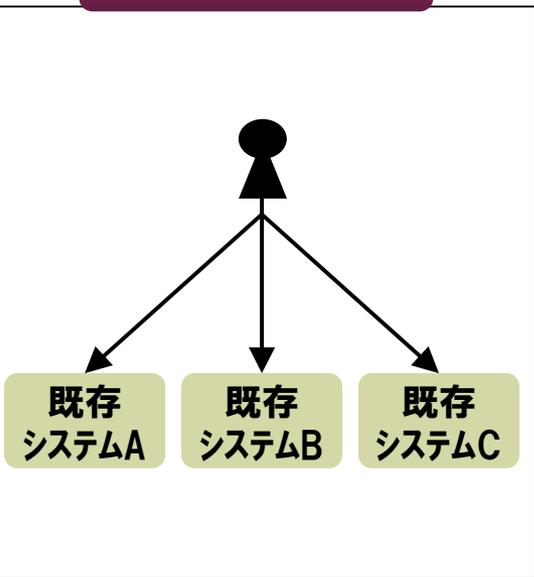
課題

- 一人の作業者が、複数業務システムを手により個々に使い分けているため、手間がかかり業務効率が悪い。
- 手作業によるミスが多発している。

実現の 考え方

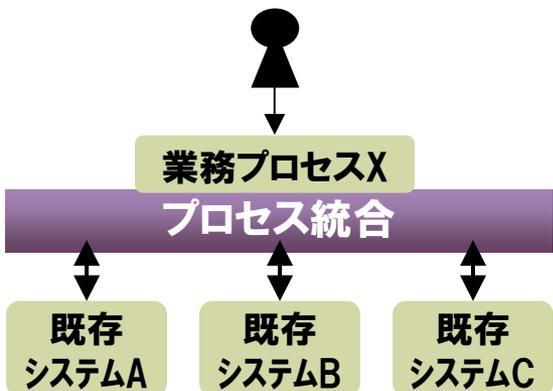
BPELにより、既存システムを組み合わせ、フロント業務でのシステム操作をワンオペレーション化し、作業者に対するワンストップサービスを実現
⇒業務プロセス効率化を早期に実現。段階的に対象を拡大

AsIs



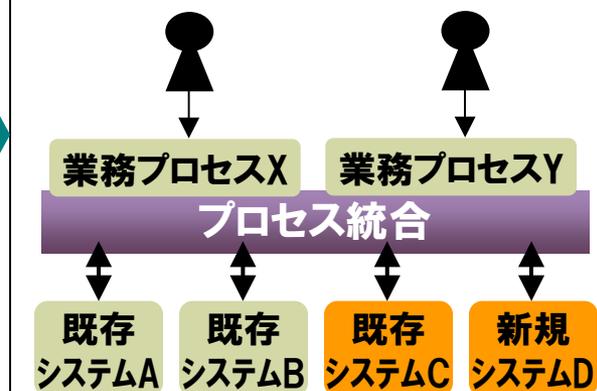
システムパターン概要

既存システムを組み合わせ、ワンストップサービスを実現 (業務プロセスをシステム化)



段階的システム最適化への適用

- ・別のワンストップサービスを追加
- ・機能拡張のため、新規システム追加や既存システム変更

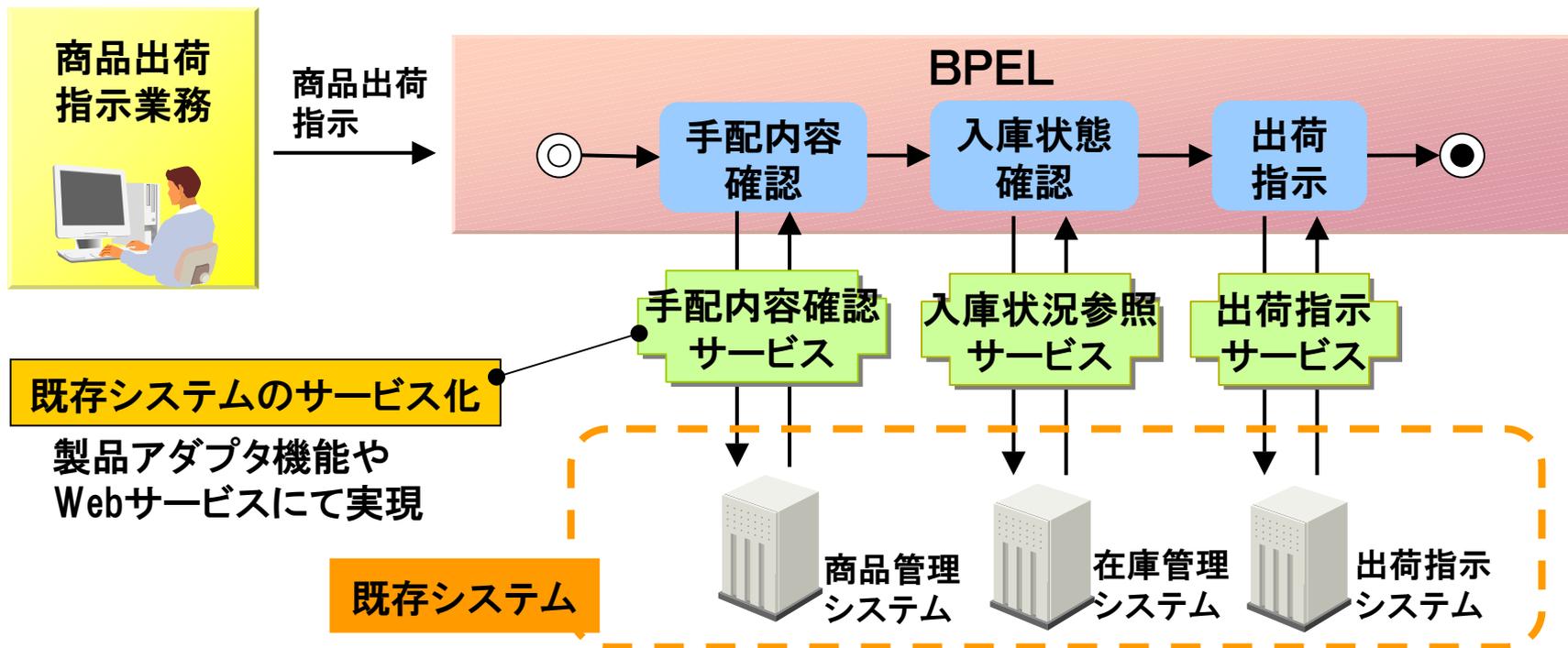


3.1-2

ワンストップサービスパターン: 適用の考え方

狙い

- システム操作のワンオペレーション化による業務の効率化を実現
- 既存システム活用による新規業務構築の迅速化を実現



メリット

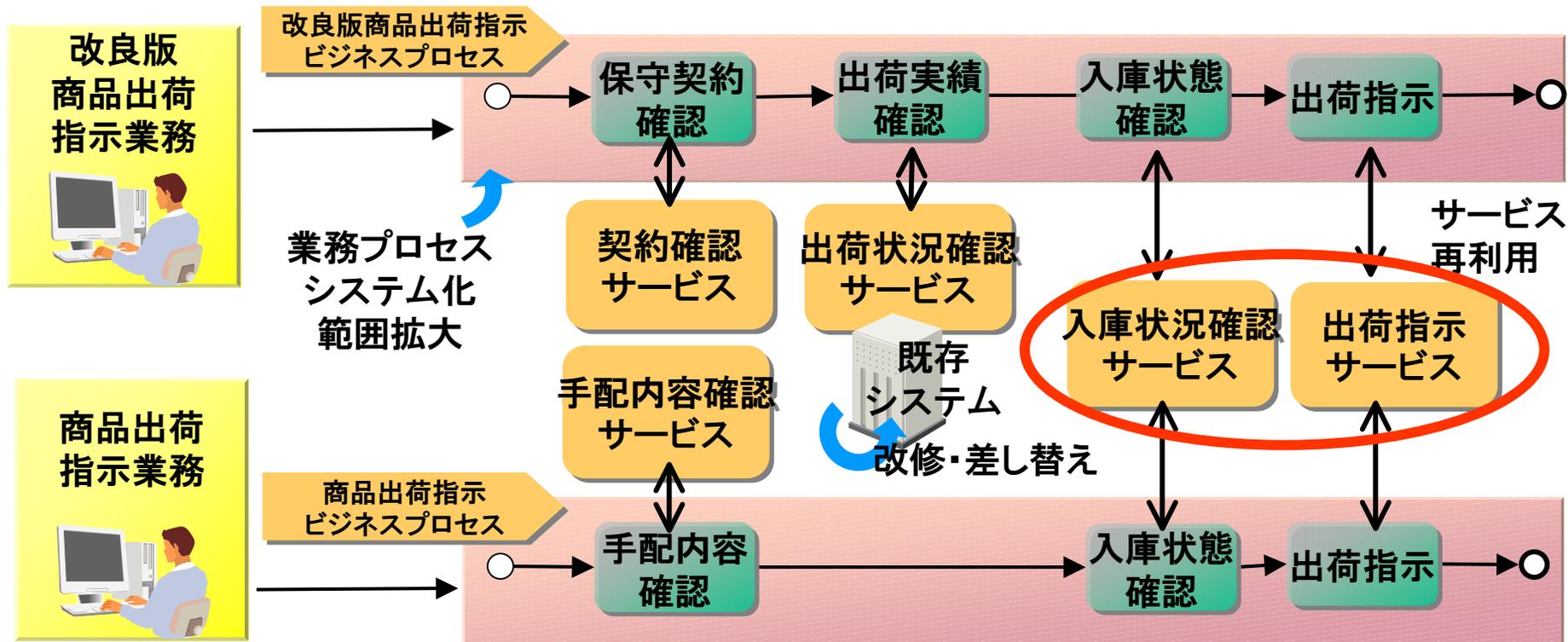
- 情報システム最適化計画の中で、業務効率化の早期実現
- 既存システム活用による工期短縮

3.1-3

ワンストップサービスパターン: 段階的システム最適化への適用

狙い

- ワンストップサービス導入による業務プロセス効率化を、他の業務にも適用
- 機能拡張のためのサービス追加や、既存システムの改修・差替え



メリット

- 類似業務のシステム化では、サービス再利用により工期短縮を実現
- 業務プロセスフロー拡張、サービスの実装変更が容易

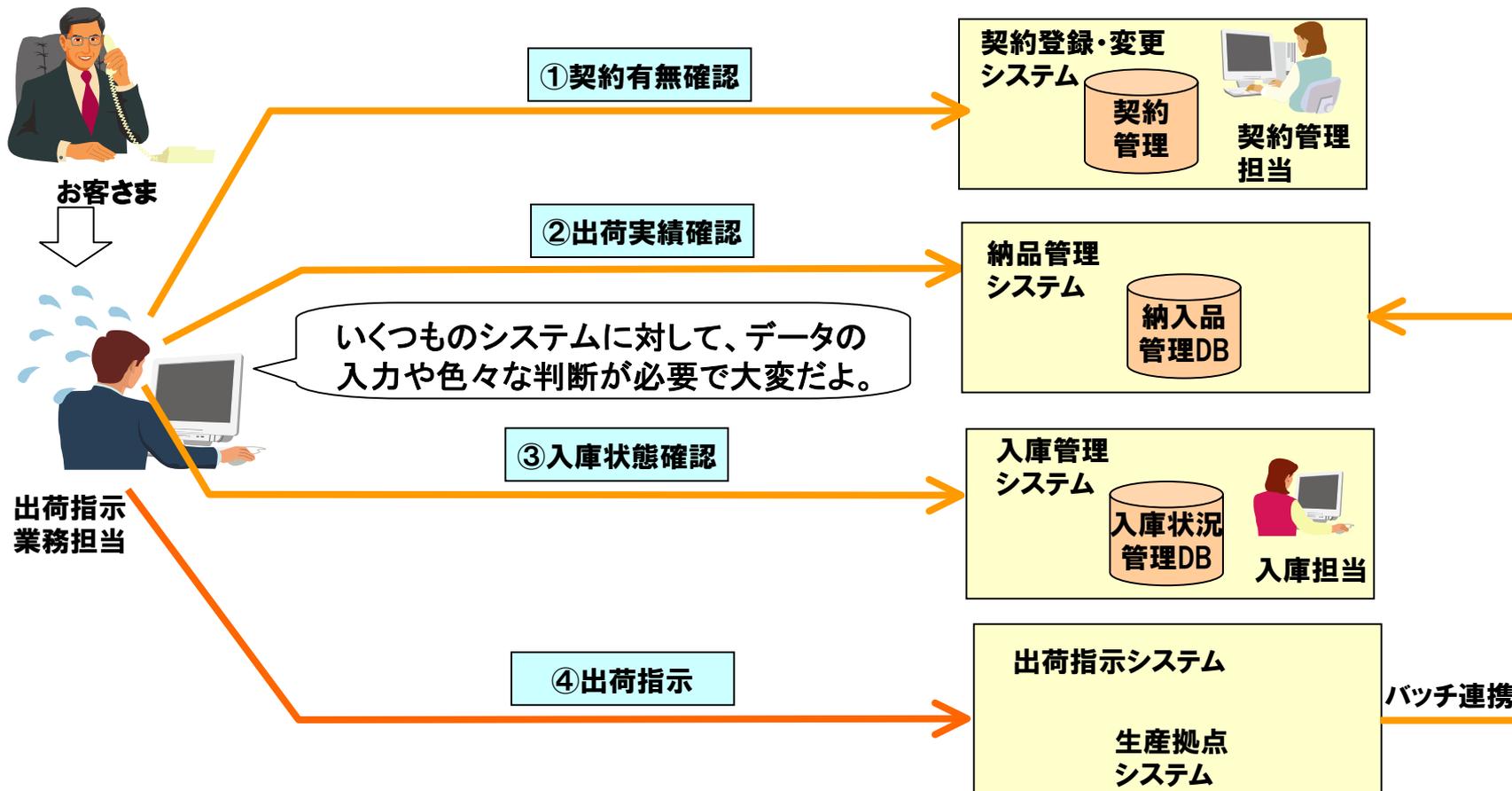
3.1-4

ワンストップサービスパターン:

【適用例】課題: 商品出荷指示業務の効率化

適用前

個別業務のシステム化は行われているが、商品出荷指示業務としては存在しない。
=> 複数のシステムを状況に応じて、使い分けなければならない。



3.1-5

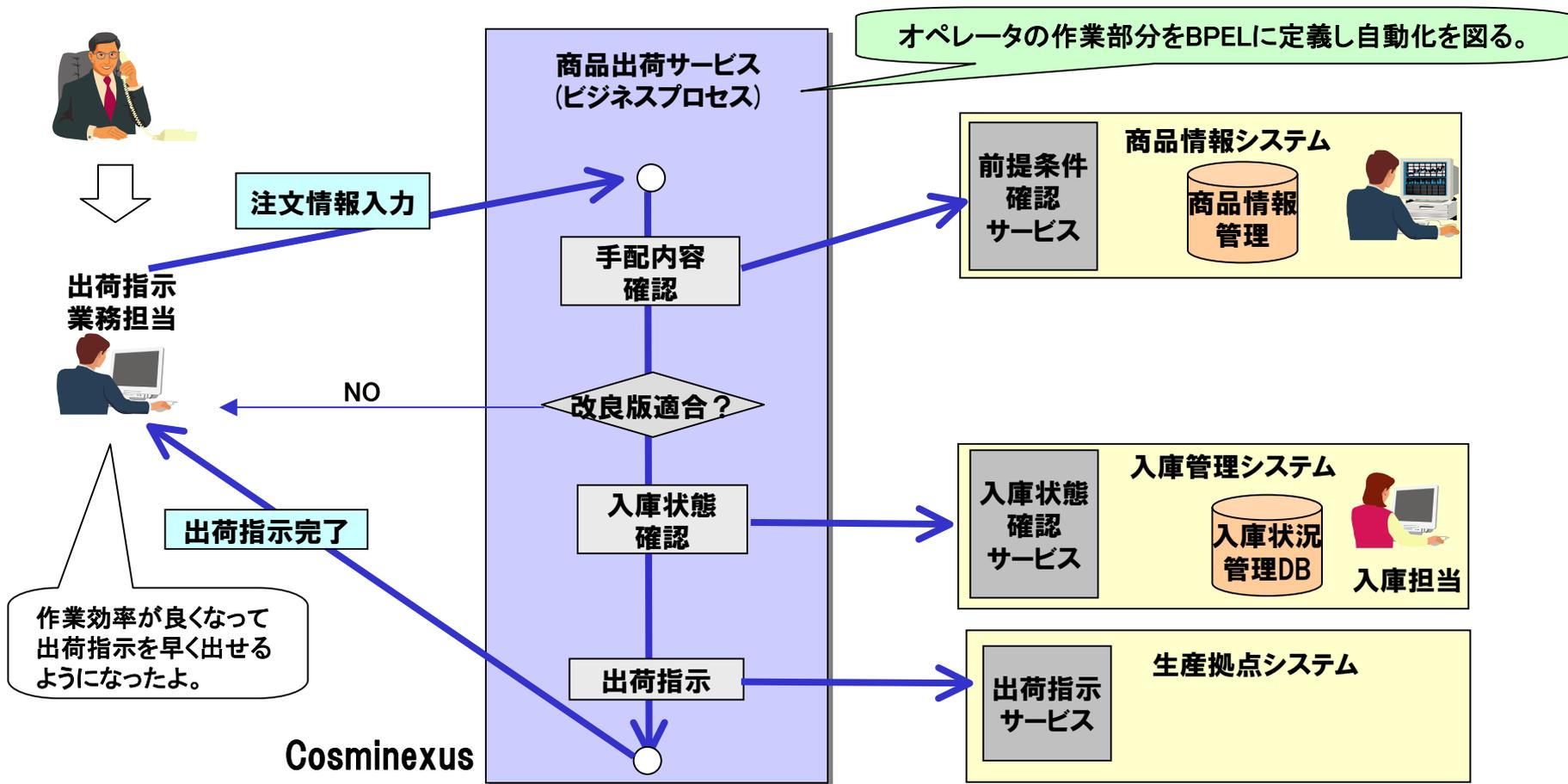
ワンストップサービスパターン:

【適用例】フェーズ1: 商品出荷指示業務のシステム化

適用後

既存システムをサービス化して組み合わせて、出荷指示業務をワンオペレーション化

- ・必要なデータは、BPEL経由で授受可能
- ・必要に応じ、UI表示・担当者入力を介在することも可能



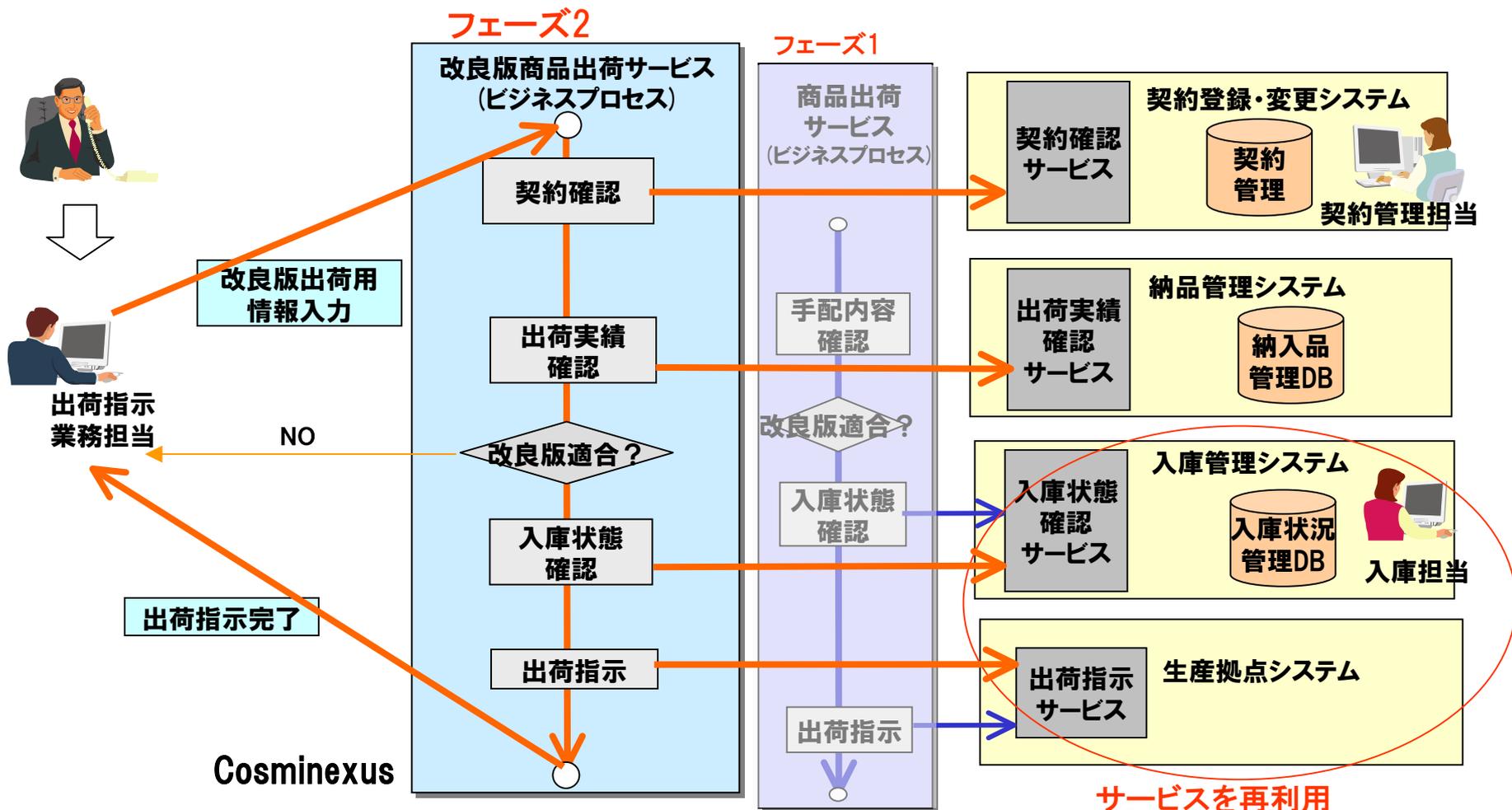
3.1-6

ワンストップサービスパターン:

【適用例】フェーズ2:商品出荷指示業務の更なる改善

改善後

他の業務プロセス(改良版製品の出荷業務)のシステム化を実施
・サービスを再利用して工期短縮



サービスを再利用

uVALUE

3.2

ビジネスプロセス自動化パターンの適用

3.2-1

ビジネスプロセス自動化パターン： 課題と実現の考え方

課題

業務システム間の連携を人手に頼っていて、非効率。
システム化して、自動化したい。

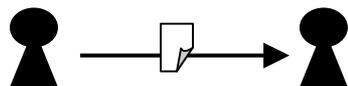
実現の 考え方

BPELにより、業務間の連携をシステム化・自動化
ファイルやDBなどの既存の仕掛けを活用して、工期を短縮
⇒業務プロセス効率化を早期に実現。段階的に対象を拡大

AsIs

システムパターン概要

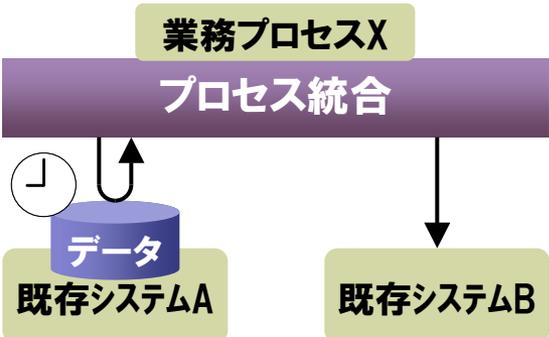
段階的システム最適化 への適用



既存システムA

既存システムB

- ・既存システムを組み合わせ、業務間連携の自動化を実現（業務プロセスをシステム化）
- ・データ到着や時刻等のイベントを契機にした連携処理の実現



業務プロセスX

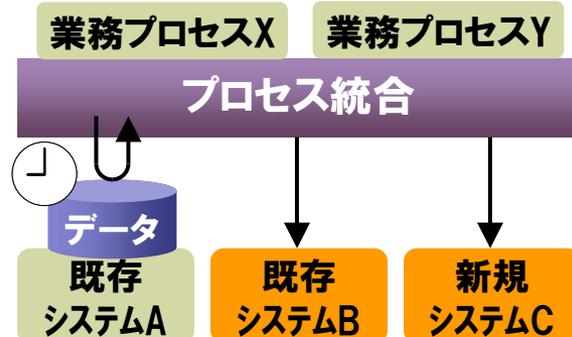
プロセス統合

データ

既存システムA

既存システムB

- ・他の業務プロセス自動化を実現
- ・機能拡張のため、新規システム追加や既存システムの刷新



業務プロセスX

業務プロセスY

プロセス統合

データ

既存システムA

既存システムB

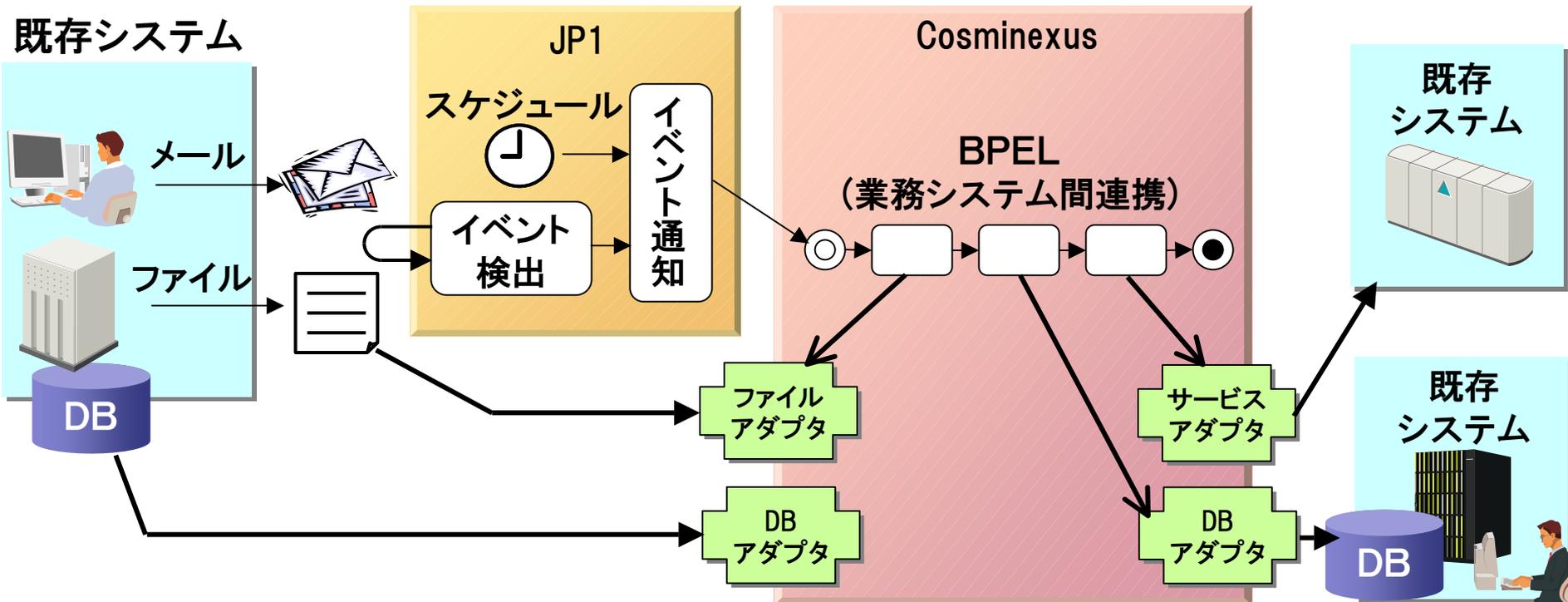
新規システムC

3.2-2

ビジネスプロセス自動化パターン： 適用の考え方

ポイント

- 複数の業務システム間連携の自動化を実現
- 既存の連携の仕掛けの再利用により、システム改修を極小化
 - ファイル生成や時刻などを契機(イベント)として連携処理を駆動
 - データ連携(ファイル連携・DB連携)を併用
- 業務プロセスフローの拡張性により、業務機能追加、既存システムの改修・差替えなど、段階的な拡張が容易



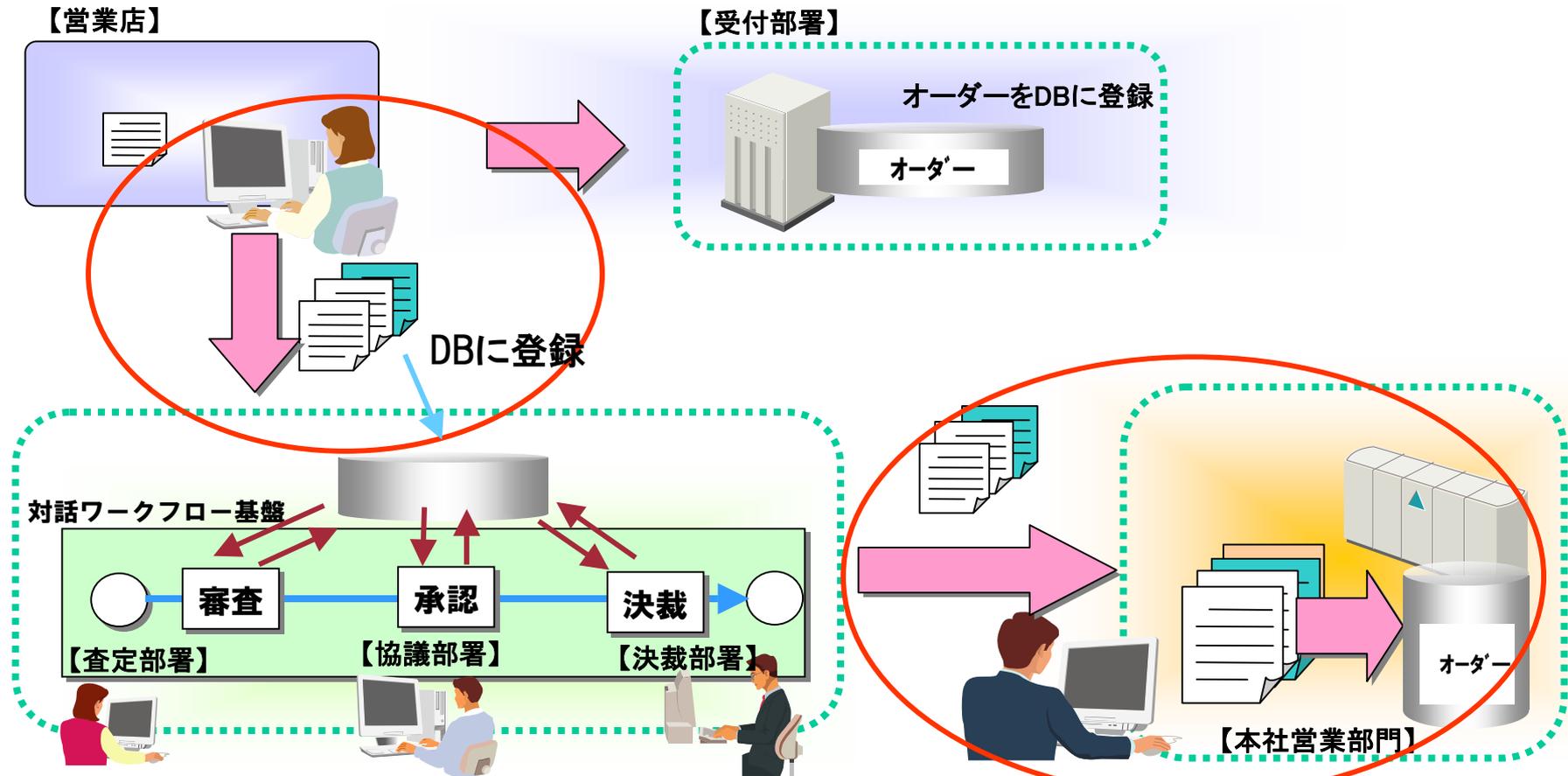
3.2-3

ビジネスプロセス自動化パターン： 【適用例】課題：受付エントリ業務の効率化

適用前

各部門間のデータ連携を人手に頼っている。

- ・各部署のDBへのデータ登録は1回/日
- ・作業効率が悪い。

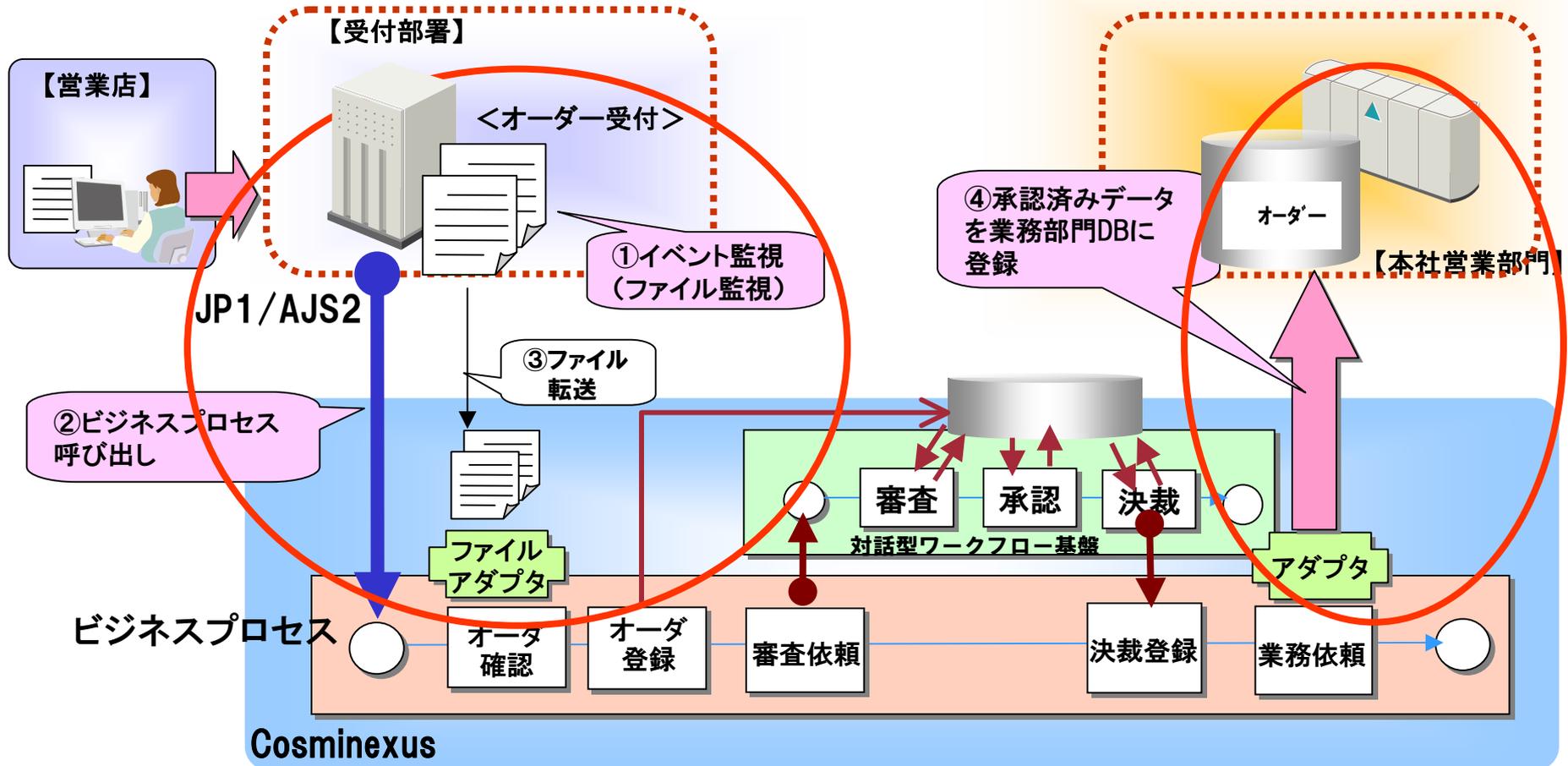


3.2-4

ビジネスプロセス自動化パターン： 【適用例】フェーズ1：受付エントリ業務の自動化

- 受付から、審査・承認、業務依頼に至る受付業務プロセス全体をシステム化
 - 受付部署でのファイル生成を契機に、受付業務プロセスを起動
- ➡ 業務プロセス最適化による業務効率向上

改善後



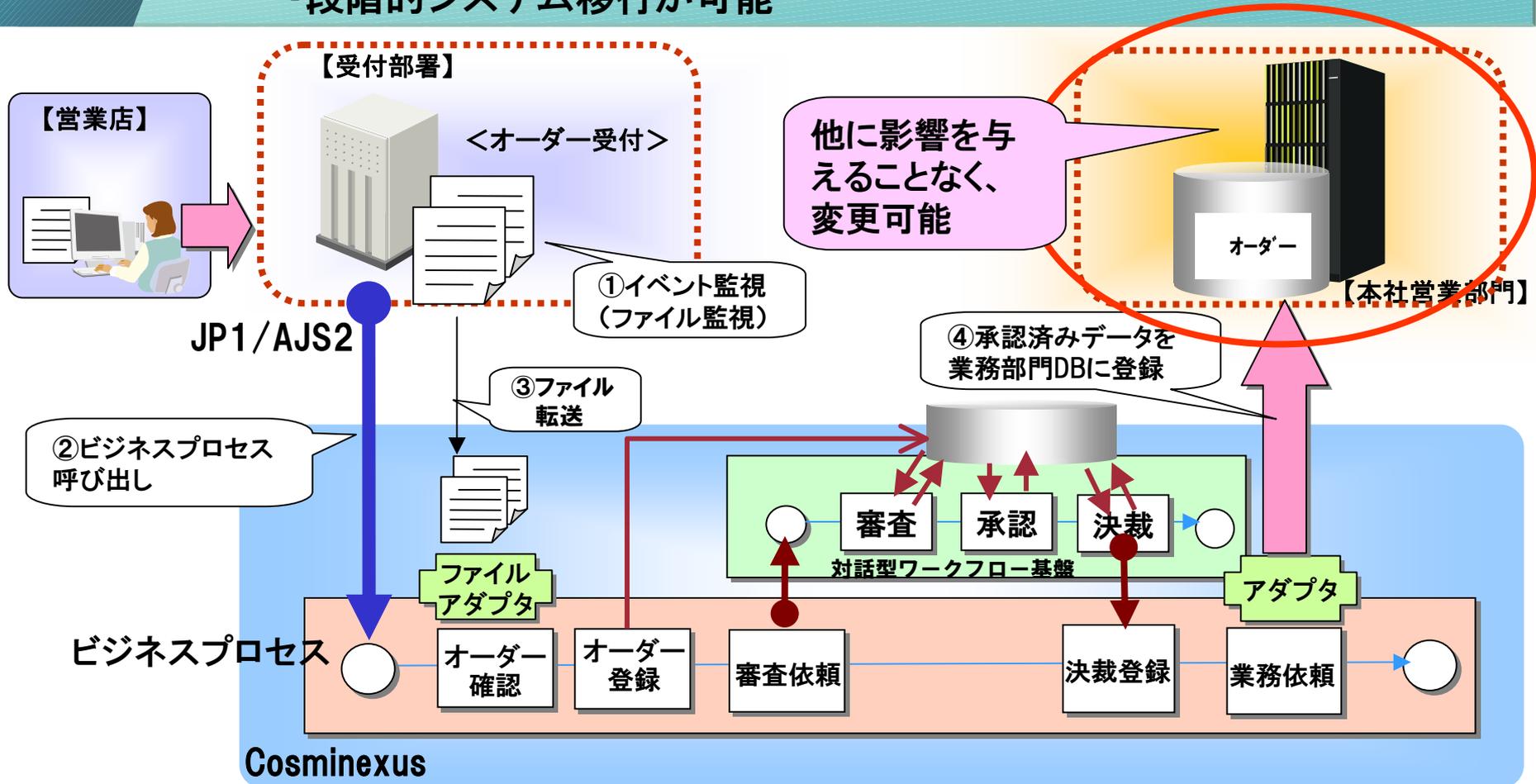
3.2-5

ビジネスプロセス自動化パターン： 【適用例】フェーズ2：既存システムの改修

営業部門のシステムを改修(例：メインフレームのオープン化)

改善後

- ・アダプタの変更のみで、他のシステムに影響なし
- ・段階的システム移行が可能



uVALUE

3.3

対話ワークフローパターンの適用

3.3-1

対話ワークフローパターン： 課題と実現の考え方

課題

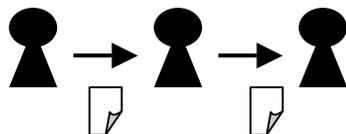
- 作業員間の連携が、人手での連携になっていて業務効率が悪い。
- 作業員間の業務の流れがルール化されていない。
- 進捗管理が不十分で、例外時の対応など業務効率が悪い。

実現の 考え方

- ワークフロー業務に適切なシステム化による業務効率の向上
 - 業務の流れのルール標準化によるコンプライアンス強化
 - 進捗管理の実現による業務効率の向上
- ⇒ 個別業務課題を早期に解決。段階的に業務プロセス最適化へ

AsIs

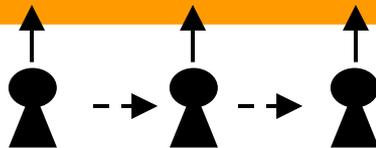
ワークフロー業務
(作業員間の連携による業務)



システムパターン概要

ワークフロー業務をシステム化

対話ワークフローシステム



段階的システム最適化 への適用

- ・業務プロセスシステム化を実施
ワークフローシステムをサービス化
- ・ワークフローから他システムに連携
- ・ワークフローの適用業務を拡大

業務プロセスX

プロセス統合

既存
システムA

既存
システムB

既存
システムC

ワークフロー
システム

ワークフロー
システム

3.3-2

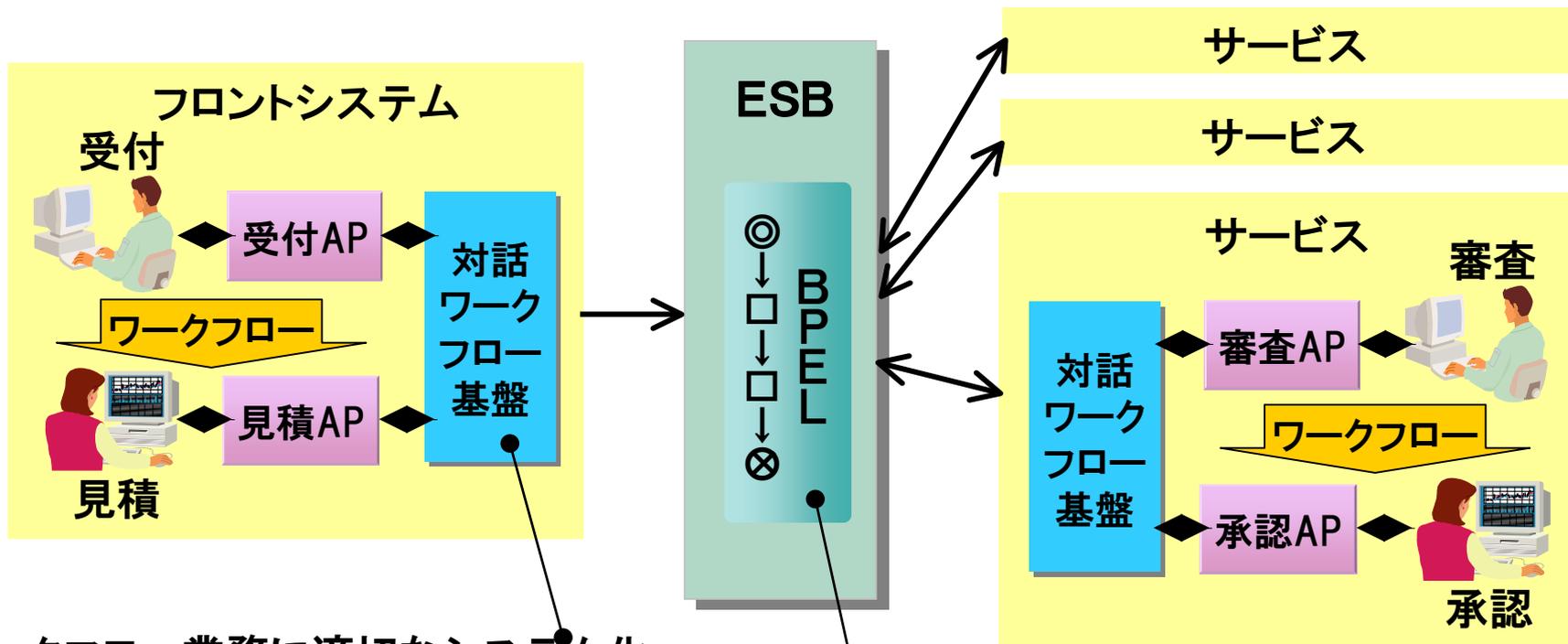
対話ワークフローパターン： 適用の考え方

フロントシステムやサービス内でのワークフロー業務の実現を先行して実施

● 業務システム内でのワークフロー業務の特有要件に適切なシステム化 (BPELの得意分野でない)

● 段階的に、対話ワークフローで構築したシステムをBPELで他システムと連携させる。

● ワンストップサービスパターン・ビジネスプロセス自動化パターンとの組み合わせによる段階的拡張



ワークフロー業務に適切なシステム化
INBOX管理、差し戻し、グループ割り当て等の特有機能要件あり。

業務プロセス・システム間連携に適切なシステム化
複数サービス連携のワンストップ化やプロセス自動化、メッセージ編集変換を伴うサービスアクセス

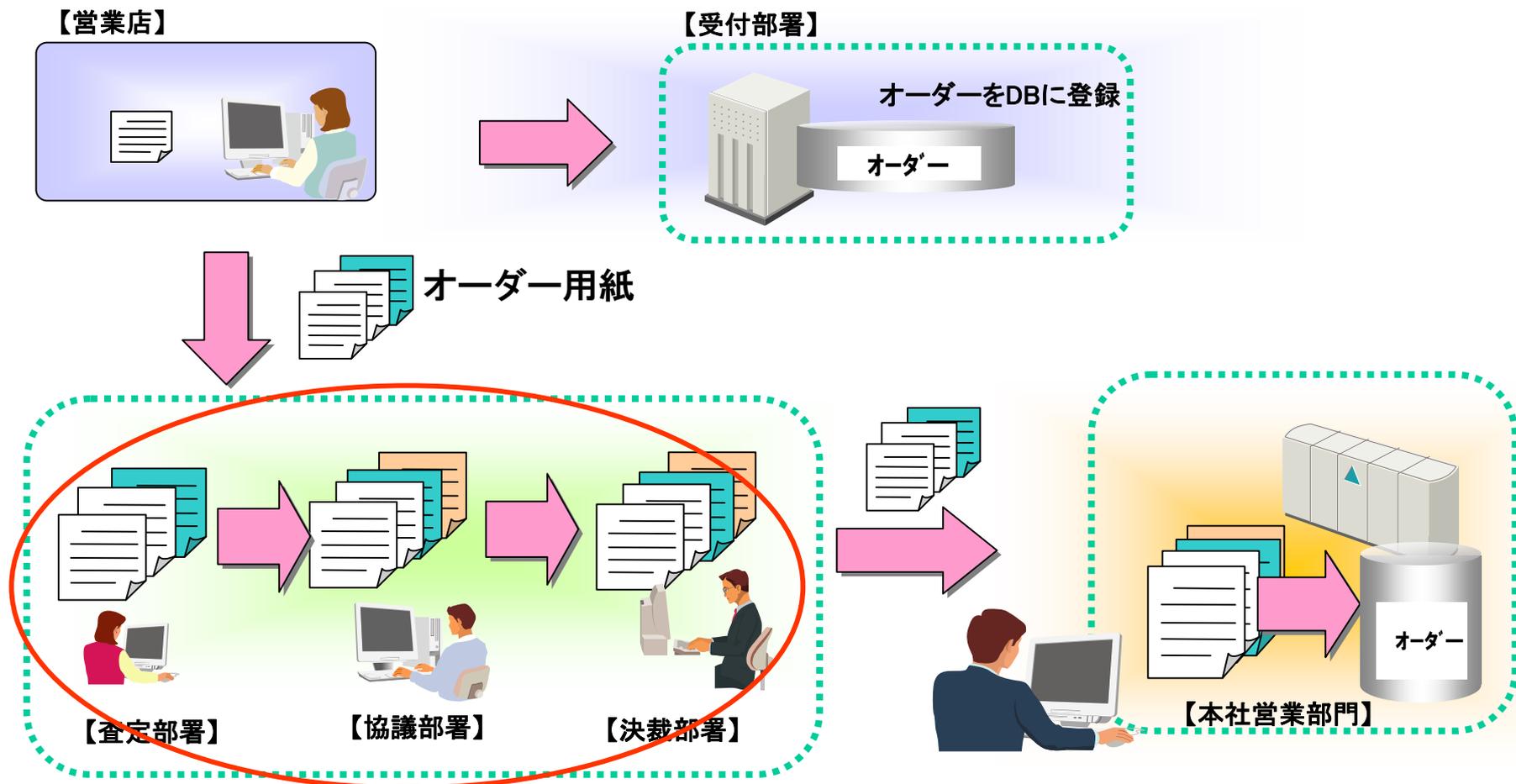
3.3-3

対話ワークフローパターン:

【適用例】課題: 受付業務の効率化

適用前

- 複数部門を横断して書類授受による業務連携になっており、業務効率が悪い。
- 特に、審査・承認業務は人手による回覧で、低効率や進捗の把握が課題



3.3-4

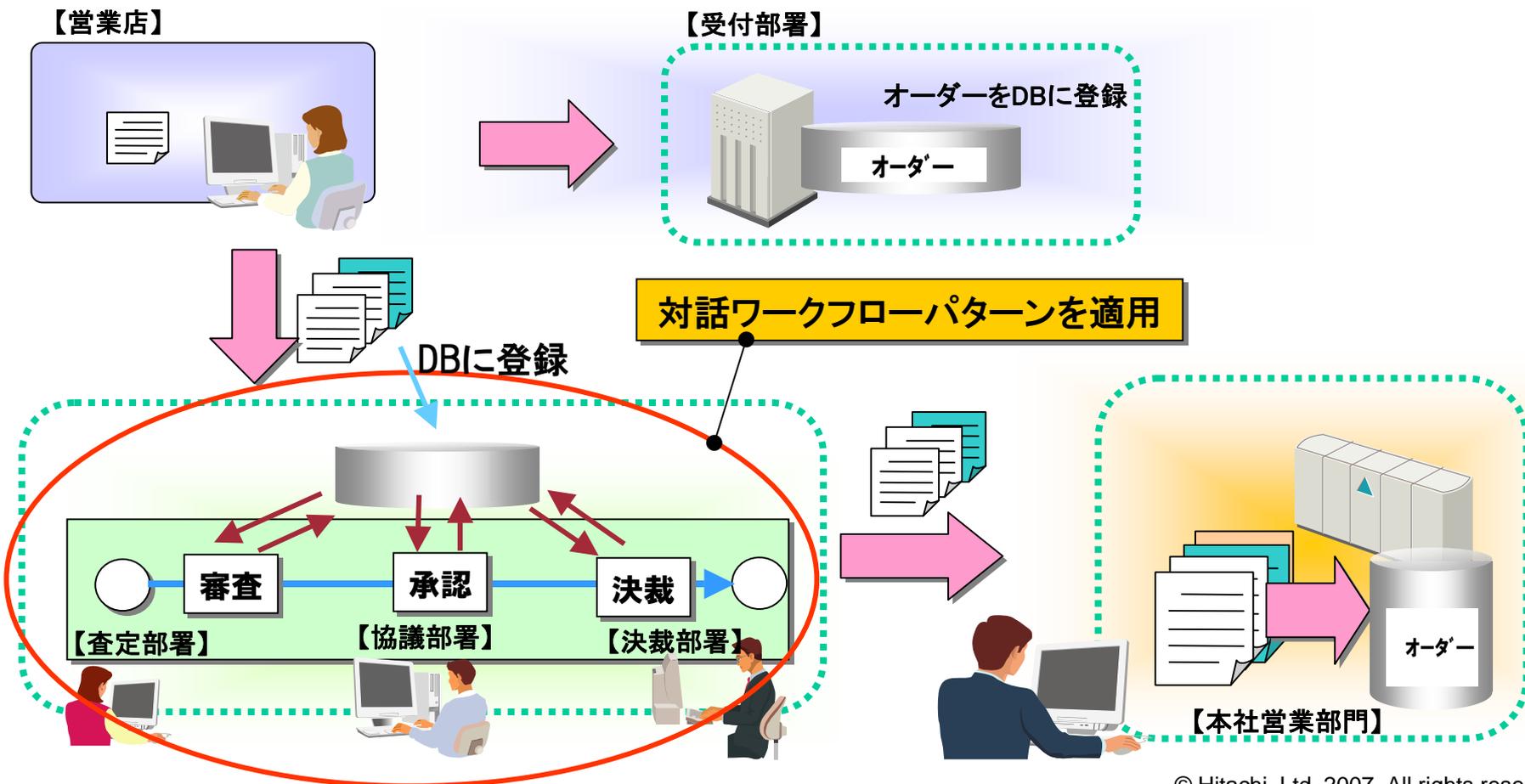
対話ワークフローパターン:

【適用例】フェーズ1: 審査・承認業務をシステム化

対話ワークフローを適用して、審査・承認の回覧処理をシステム化

適用後

- ・審査、承認処理の効率向上
- ・進捗状況の管理が可能



3.3-5

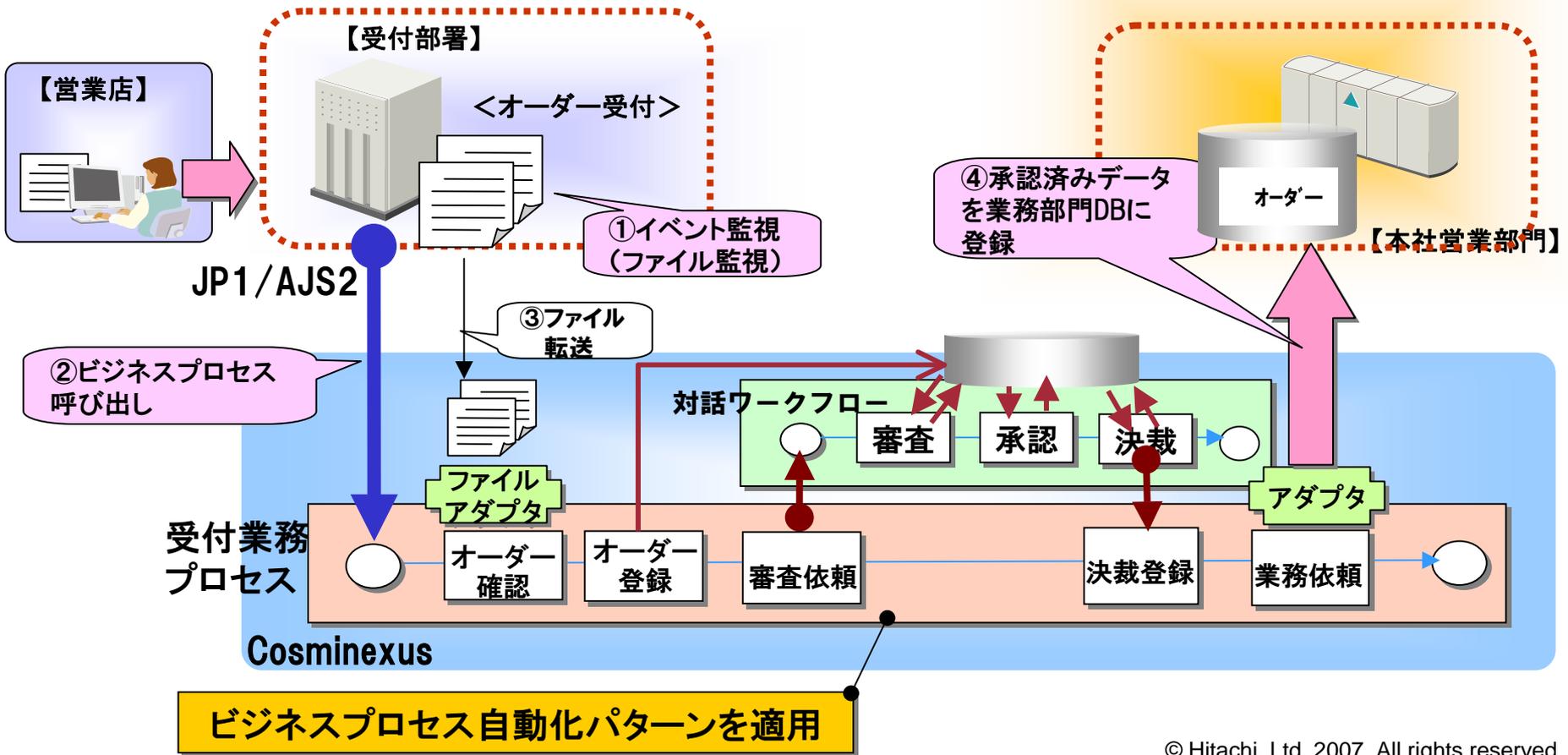
対話ワークフローパターン:

【適用例】フェーズ2: 受付業務プロセスの自動化

改善後

- 受付から、審査・承認、業務依頼に至る受付業務プロセス全体をシステム化
 - 受付部署でのファイル生成を契機にして、受付業務プロセス実現を容易化
- ➡ 業務プロセス最適化による業務効率向上

対話ワークフローパターンとビジネスプロセス自動化パターンの組み合わせにより、段階的拡張を実現



uVALUE

4

統合システム構築基盤Cosminexusのご紹介

統合システム構築基盤Cosminexus

拡張性・生産性・運用性を向上するシステム構築基盤

- システム構築の羅針盤となるリファレンスアーキテクチャや適用ガイドを提供
- 多様な形態でのサービス統合・サービス構築を実現する基盤機能



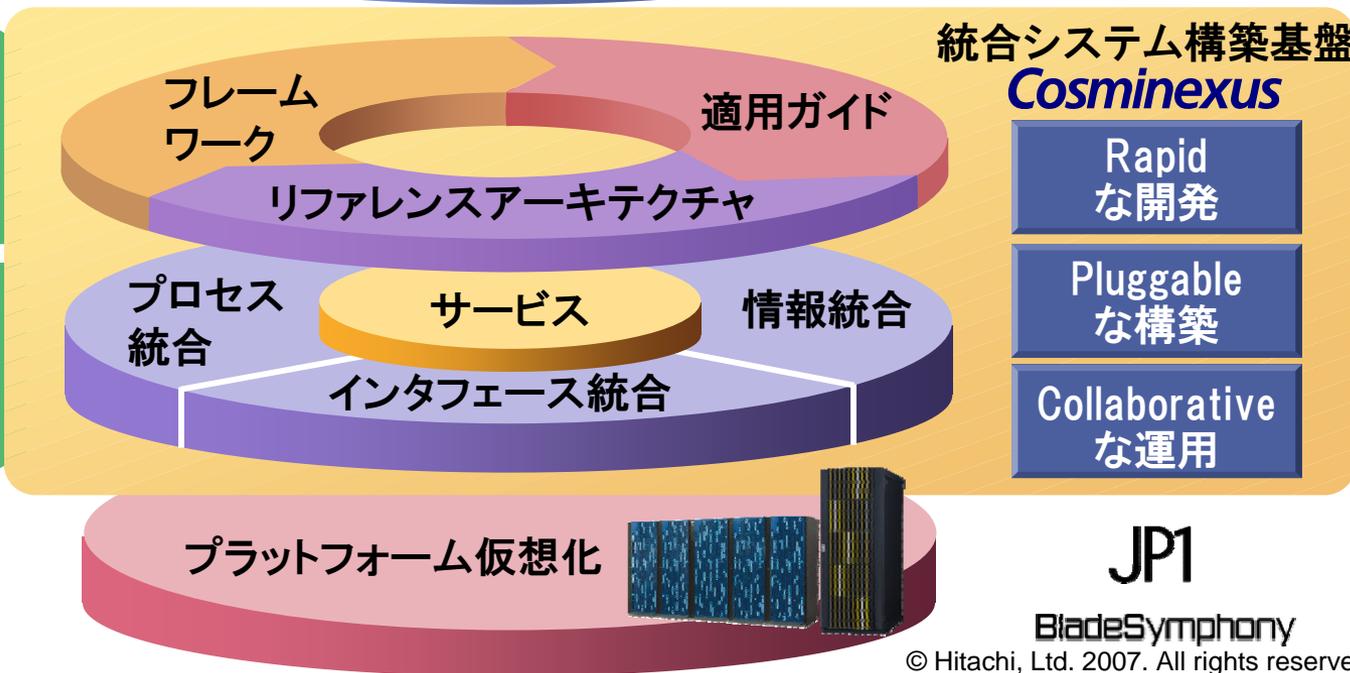
SIノウハウを
体系化したガイドや
リファレンス

変化に強く、
高い生産性を実現
できる基盤製品

Cosminexus

OpenTP1

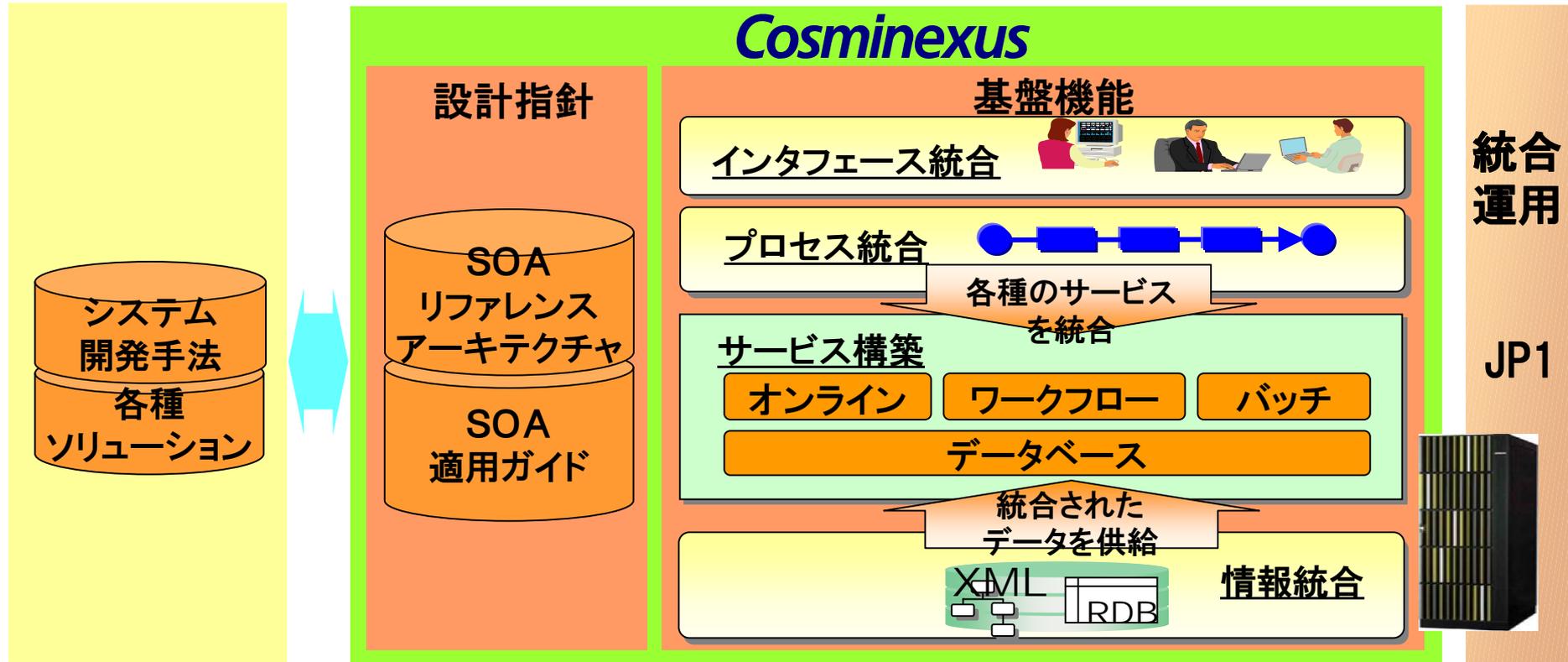
HiRDB



4-2

Cosminexusの特長

- 多様なサービス構築・サービス統合において、生産性・拡張性・運用性を向上
 - オンラインに加え、対話ワークフロー、バッチといった多様な処理形態のサービス構築を実現
 - インタフェース・プロセス・情報の統合で、業務プロセスの効率化とシステム拡張性を向上
- 設計・構築を効率化するガイドラインドキュメントを提供し、生産性を向上
 - 顧客課題に直結したシステムパターン定義を中心に設計指針を提供
 - システム設計の部品として、各種の開発手法やソリューションに柔軟に適用可能



END

情報システムに拡張性・生産性向上をもたらすSOA実践アプローチ

株式会社 日立製作所 ソフトウェア事業部 先端情報システム研究開発部
田坂 光伸

《謝辞》

- uCosminexus Service Platform、uCosminexus Service Architect、uCosminexus Application Server、uCosminexus Developerは、経済産業省が2003年度から3年間の予定で推進中の「ビジネスグリッドコンピューティングプロジェクト」の技術開発の中間成果を含みます。

《他社所有名称に対する表示》

- Java及びすべてのJava関連の商標及びロゴは、米国及びその他の国における米国Sun Microsystems,Inc.の商標または登録商標です。その他記載の会社名、製品名は、それぞれの会社の商標もしくは登録商標です。

uVALUE *with Harmonious* Computing*

HITACHI
Inspire the Next