

# 効果に着目したBPMの考え方と 日立が考えるクラウド/SOA適用

～ワークショップから得た正しい評価軸とSOA適用パターン～

2012/11/9

株式会社 日立製作所 情報・通信システム社  
ソフトウェア本部  
システム基盤ソリューション部

吉村 誠

Human Dreams.  
Make IT Real.

**効果に着目したBPMの考え方と日立が考えるクラウド/SOA適用  
～ワークショップから得た正しい評価軸とSOA適用パターン～**

## Contents

1. ビッグデータにも備える！ BPMの考え方と情報システム
2. SOAの適用パターンとワークショップから得た効果の評価軸
3. 「見えるクラウド」を実現するための取り組み
4. まとめ

**効果に着目したBPMの考え方と日立が考えるクラウド/SOA適用  
～ワークショップから得た正しい評価軸とSOA適用パターン～**

# **1. ビッグデータにも備える！BPMの考え方と情報システム**

## BPMとは

### ● Business Process Management

ビジネスプロセス(業務の流れ)を単位ごとに分析・整理することによって問題点を見出し、最適な作業の仕方に継続的に改善する管理手法

## ビジネスプロセスとは？

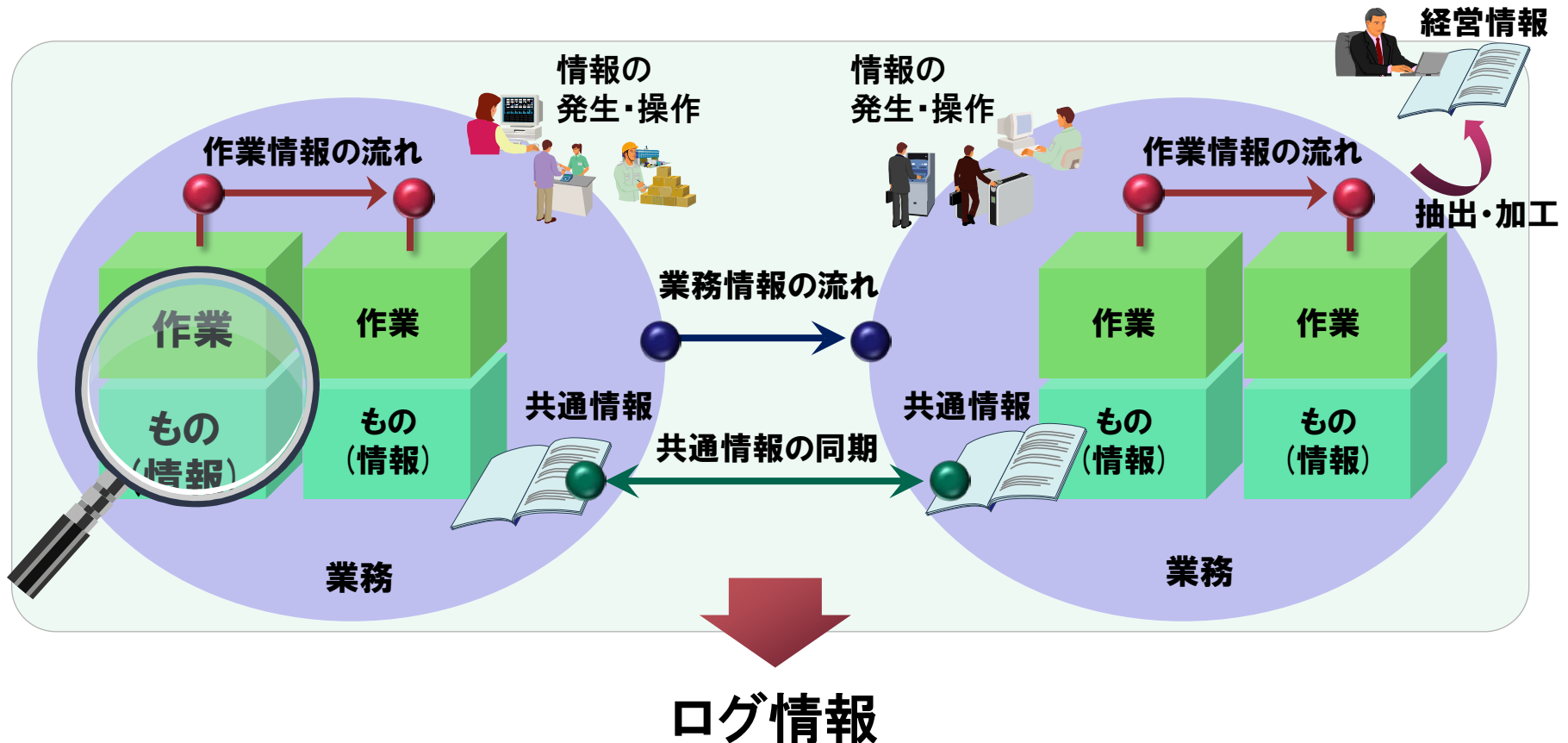


一般には、こういった単位を思い浮かべるが、それだけなのか？

# 1-2 そもそも業務処理は？(業務と作業の構造の抽象化)

## 業務処理構造は

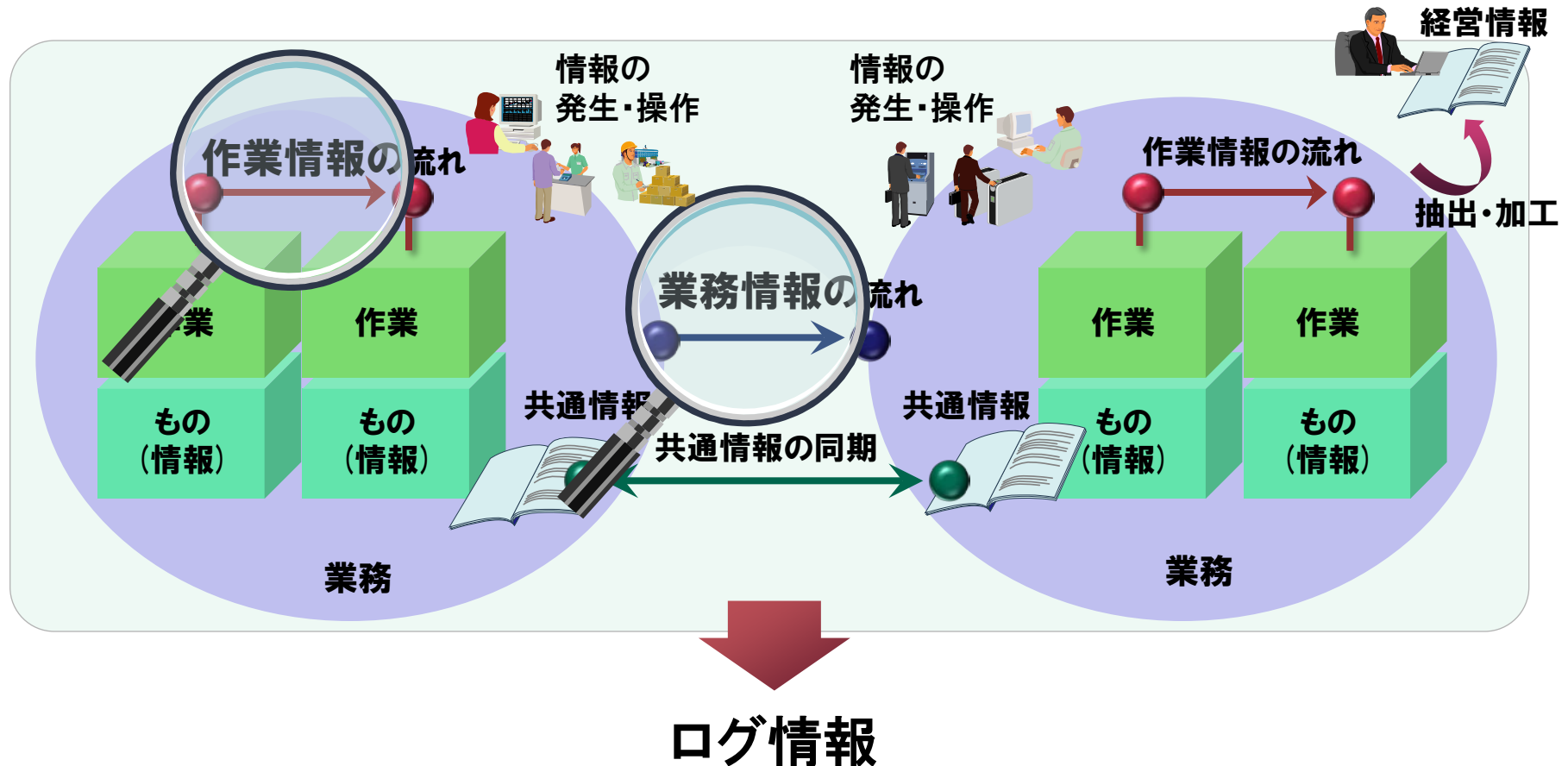
- そもそも「業務」とは、個人(あるいは機械)が実施する一つ以上の「**作業**」で構成される。
- 「作業」はそれに付随する「**もの(情報)**」を使って遂行される。



# 1-3 そもそも業務処理は？(業務と作業の構造の抽象化)

## 業務処理 構造は

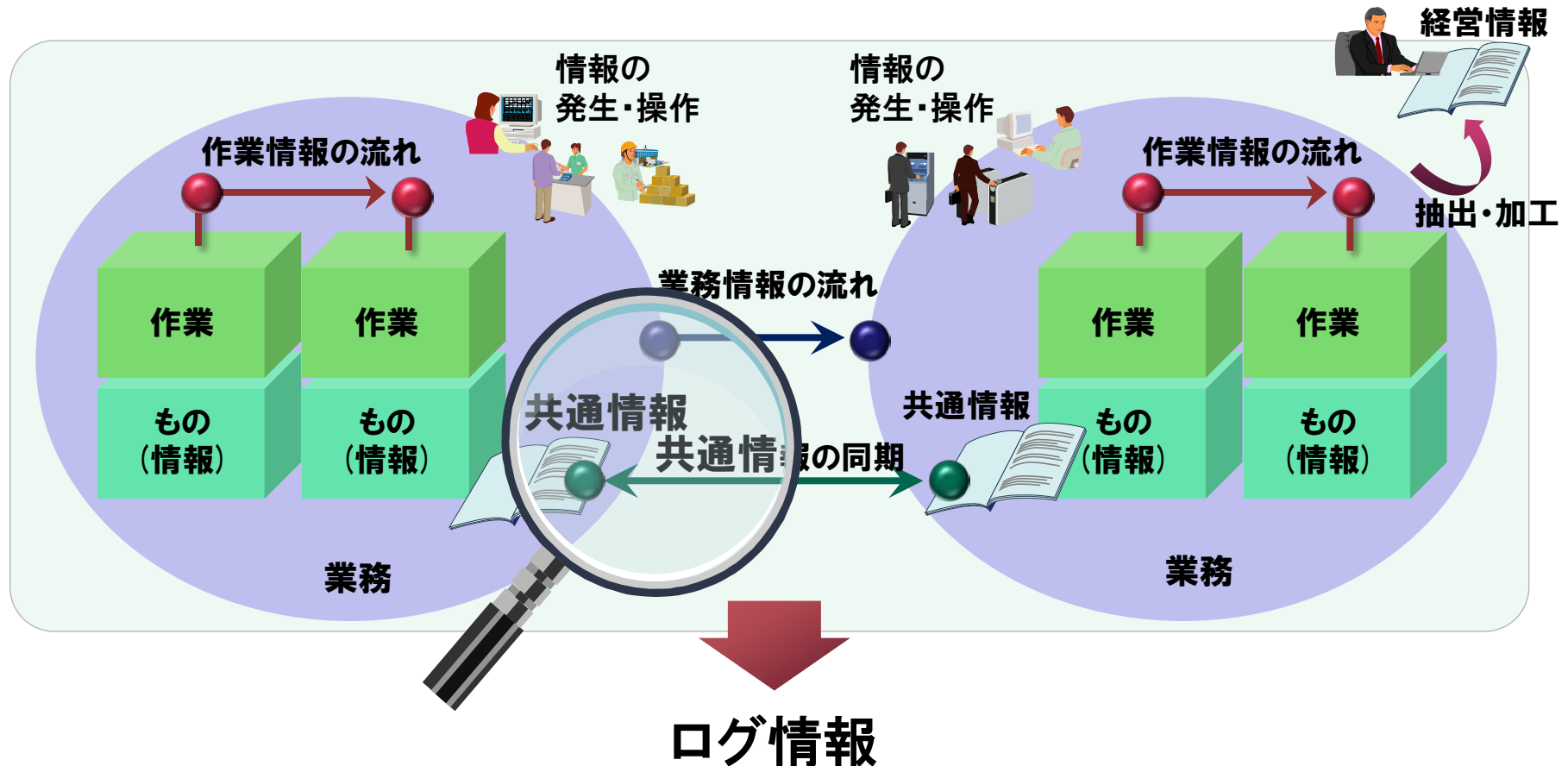
- 「作業」と「作業」の組み立てには「**作業の流れ(プロセス)**」が存在する。
- 「業務」と「業務」の間にも「**業務の流れ(プロセス)**」が存在する。



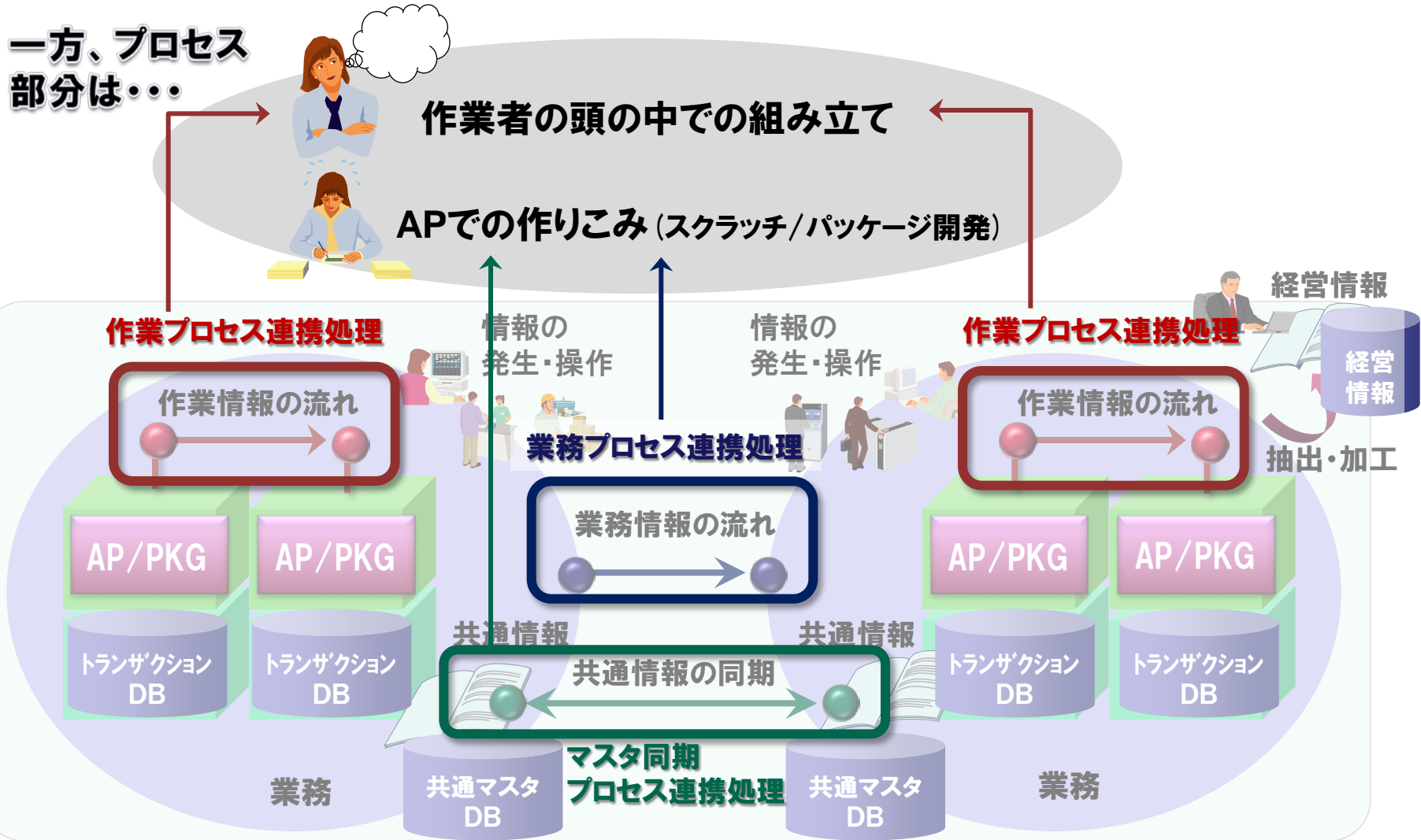
# 1-4 そもそも業務処理は？(業務と作業の構造の抽象化)

## 業務処理 構造は

- 「情報」は個々の「作業」で発生・更新される「情報」と「作業」を遂行するために参照される「**共通情報(マスタ)**」が存在する。
- 「共通情報(マスタ)」には、それを**同期させるプロセス**も存在する。



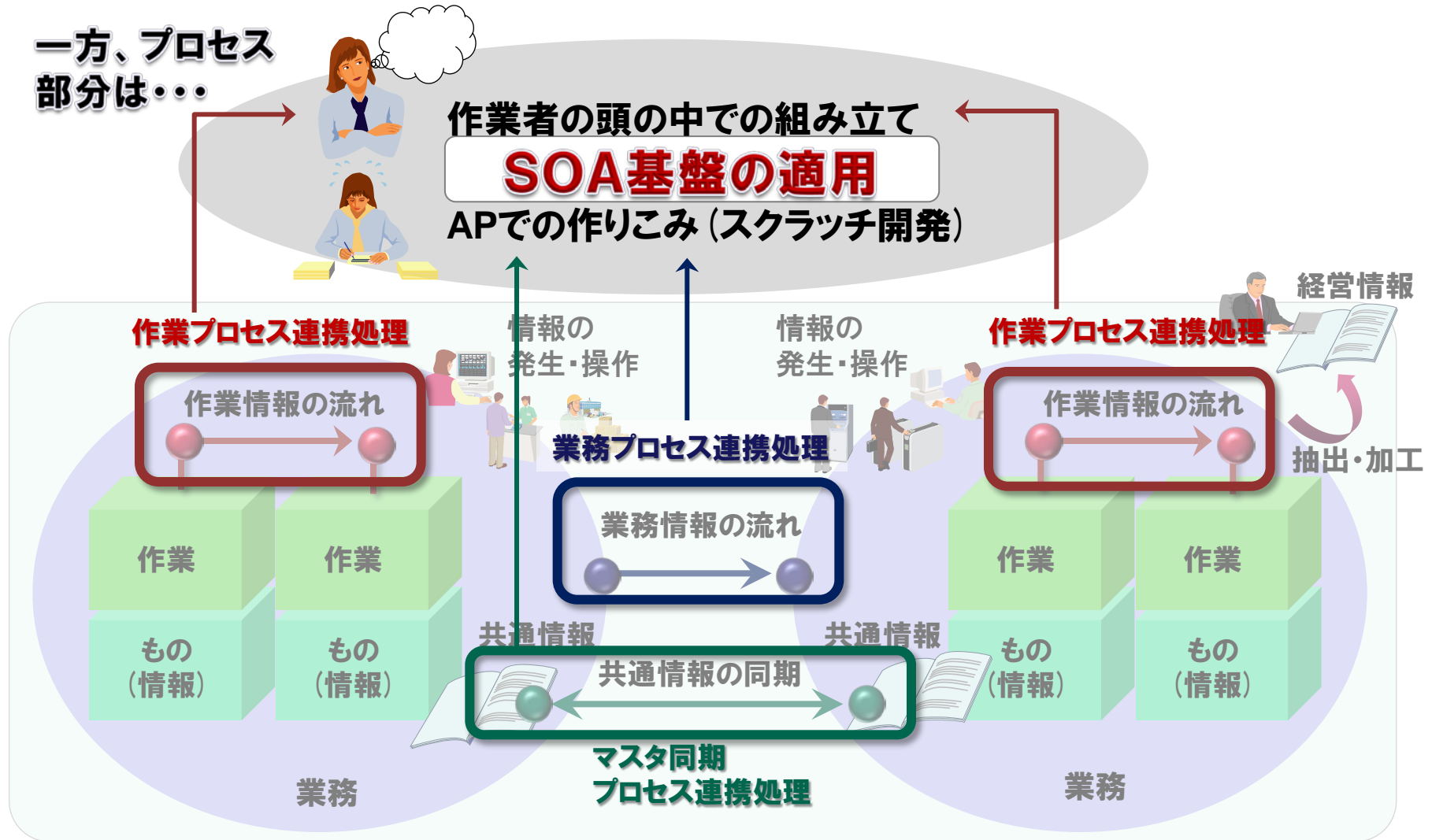
# 1-5 そもそも業務処理は？ (IT化:スクラッチ/パッケージ開発の場合)



## APやDBなど、全てスクラッチ/パッケージ開発の場合

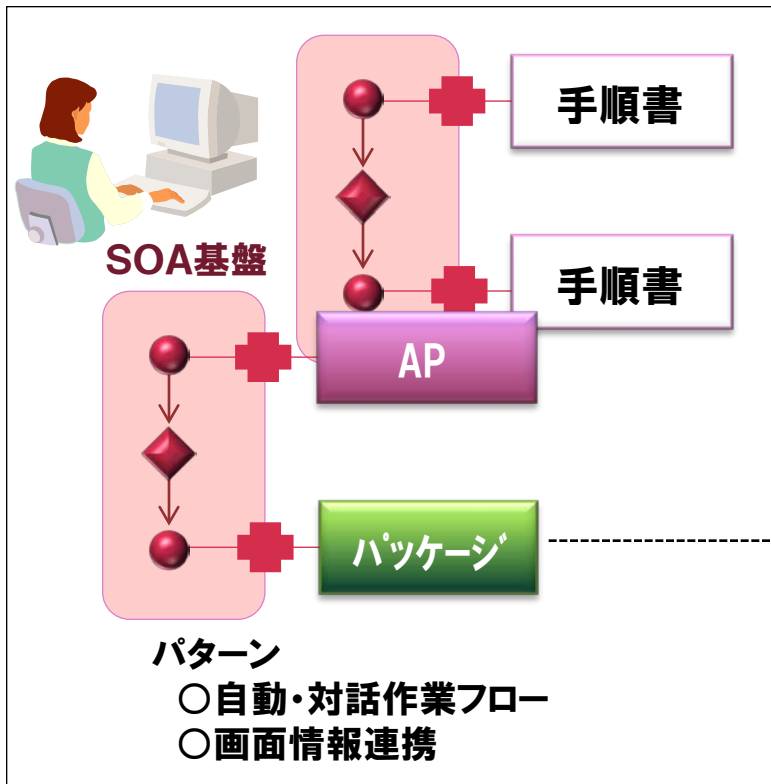


# 1-6 そもそも業務処理は？(SOA基盤の適用部分)



この「プロセス」部分が管理すべき「ビジネスプロセス」では？  
→この「プロセス」部分をIT化する基盤が「SOA基盤」

## 作業プロセス連携処理 (個人の作業連携)

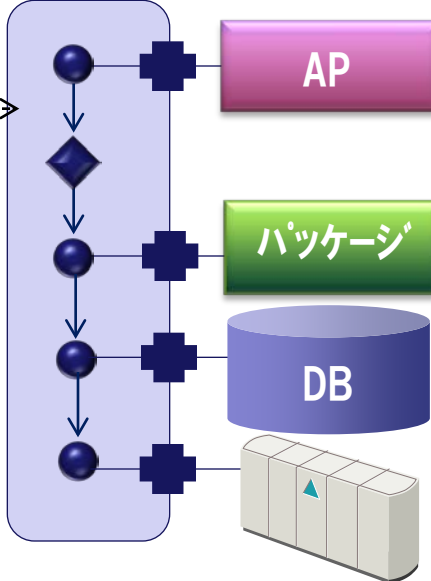


## 業務プロセス連携処理 (システム間、組織間、個人-個人)

### パターン

- 自動業務フロー (Hub型、業務フロー型)
- 対話業務フロー

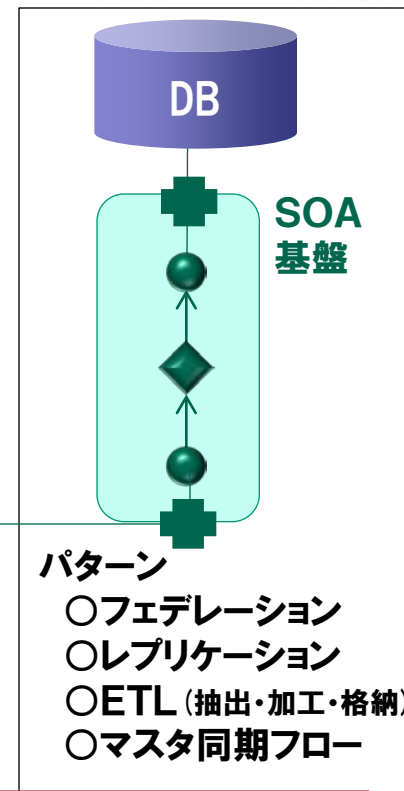
### SOA基盤



## SOA基盤の特徴

- ・プロセス定義・実行
- ・接続アダプタ提供
- ・データ変換機能
- ・ログ出力(内部統制)

## マスタ同期プロセス連携処理



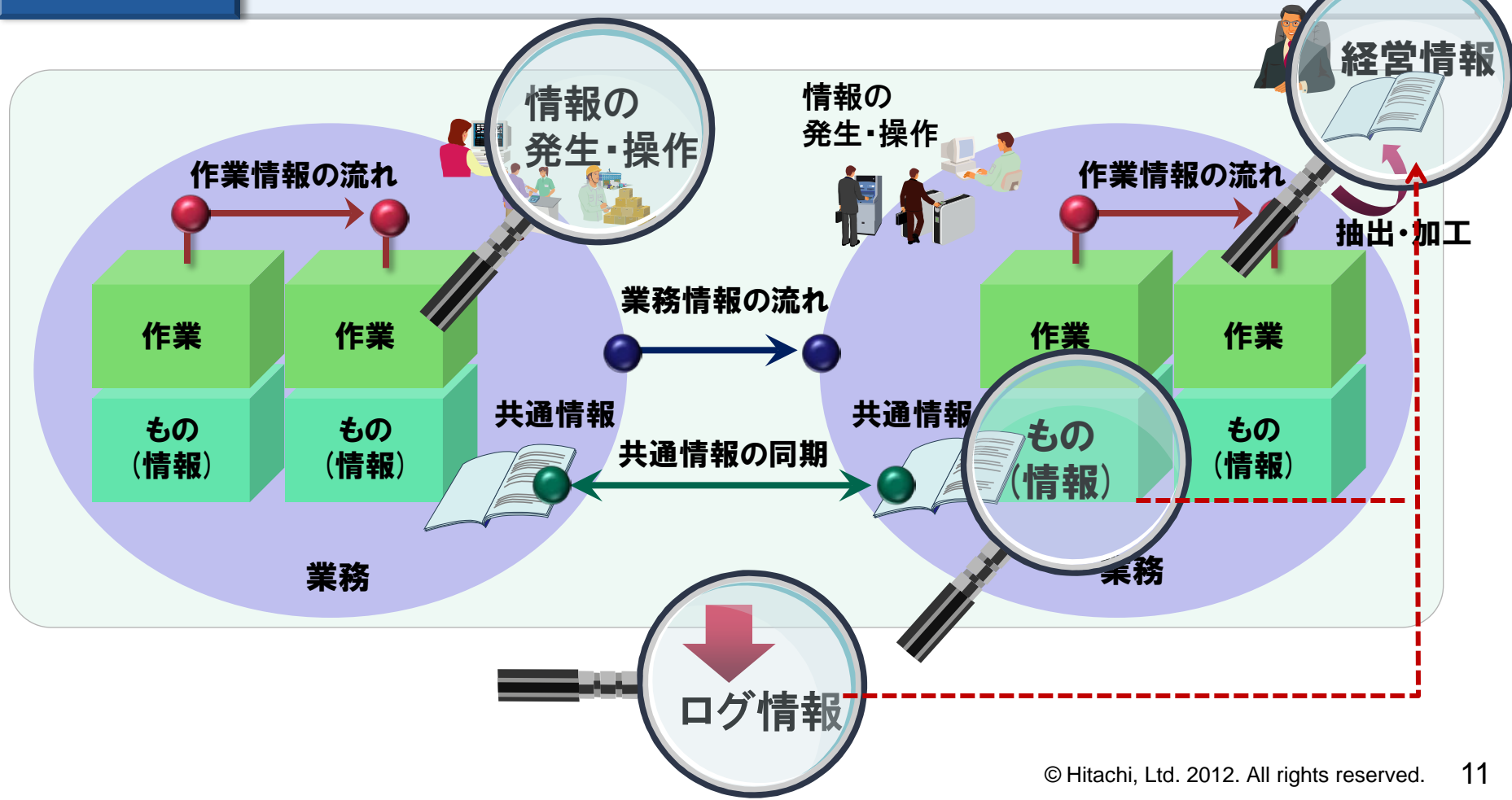
SOA基盤は 作業のプロセスからデータプロセスまで 支援する基盤



# 1-9 そもそも業務処理は？(業務と作業の構造の抽象化)

## 業務処理構造は

- 「情報」の発生源は人・機械。
- 同一情報が複数箇所で発生することは、あってはならない。
- 抽出・加工された情報は、経営情報として扱われる。
- 「情報」には、付随して発生する「ログ情報」も存在する。



# 1-10 「見える化」におけるBAMの考え方①

## ● BAM(Business Activity Monitoring)の目的

業務におけるKPI(処理量・処理時間・データ内容)をリアルタイムにモニタリングすることで問題の把握と対応を迅速に行う。

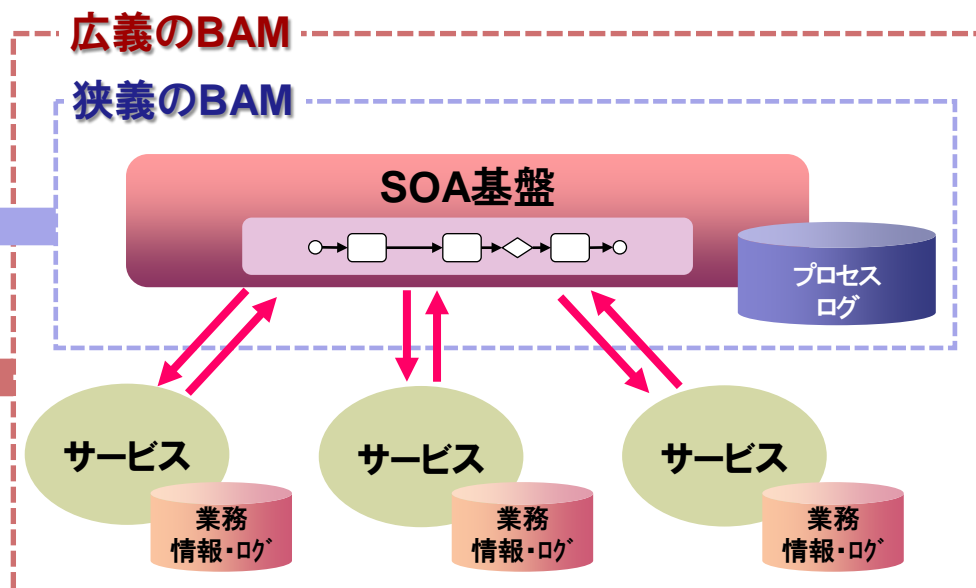
## ● 設定したKPI値をモニタリング

### 狭義のBAM

・SOA基盤が蓄積するプロセスログの監視、分析

### 広義のBAM

・SOA基盤が蓄積するプロセスログの監視、分析  
・AP処理のログと監視も含めた業務データの分析



## ● ルールに基づいてトラブル発生をいち早くキャッチ

SOA基盤により提供される機能の例:

機能	説明
ダッシュボード	モニタリングする情報を編集・表示し、業務改善やIT投資などの経営判断に活用
アラート	モニタリングする情報から不正操作やシステムの問題発生を検知し、リアルタイムに警告

# 1-11 「見える化」におけるBAMの考え方②

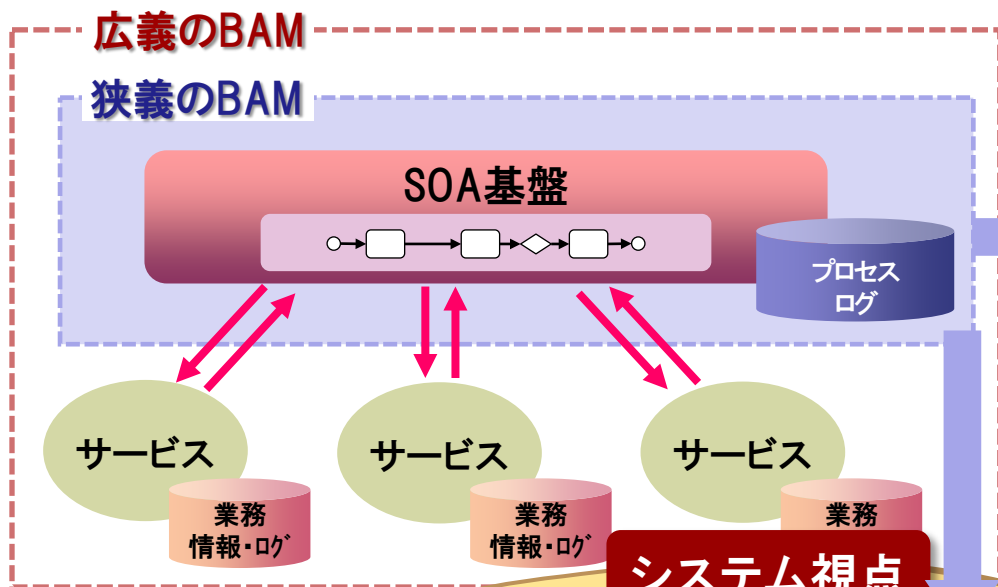
## ポイント

### ● 監視面

モニタリングする情報から、不正操作やシステムの問題発生を検知し、リアルタイムに警告することなどができる。

### ● 改善面

モニタリングする情報を編集・表示し、**業務改善**や**IT投資**などの経営判断に活用できる。



## 業務視点

業務  
主管部門

しかし、プロセスログだけでは

- ・リードタイム、ターンアラウンドタイムしか分からない。
- ・業務と業務など、全体をプロセスで繋ぐのは一気にはできないし、アプリケーションログまでを含めないと真の業務改善に繋がられない。



システム  
管理部門

でも、プロセスログで

- ・どこからどこへ、どんなデータがどういったプロトコルで、かつ、どういった頻度で渡っているのか？
- ・その際のレスポンスは？ スループットは？
- ・同じようなデータが色々なところに渡ってはいないか？

- ・まずはシステム間をHub的な接続形態で段階導入し、IT投資の観点でログを活用。
- ・情報システム部門でクローズできることなどから比較的適用が始めやすい



# 1-12 そもそも業務処理は？(IT化:経営情報の抽出・加工)

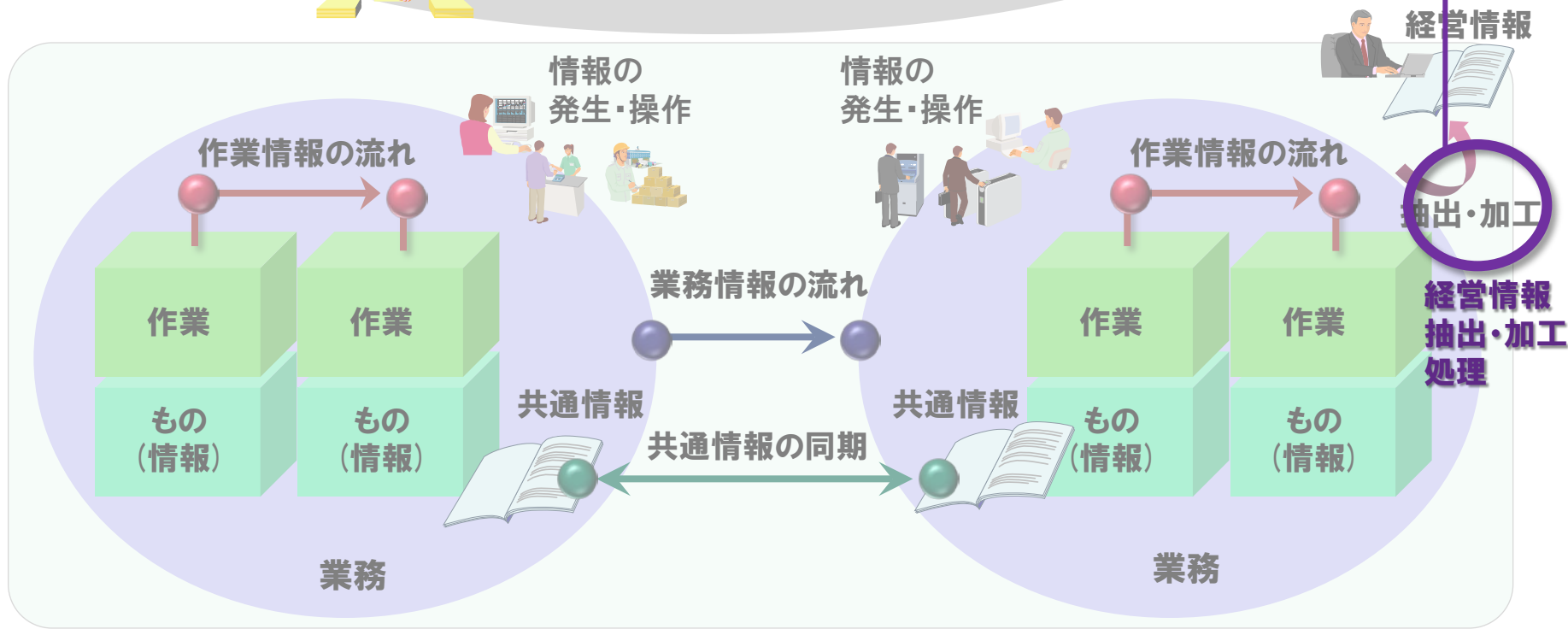
抽出・加工の  
処理部分は...



分析処理にBIツールを使うケース  
が多いが...

## SOA基盤の適用

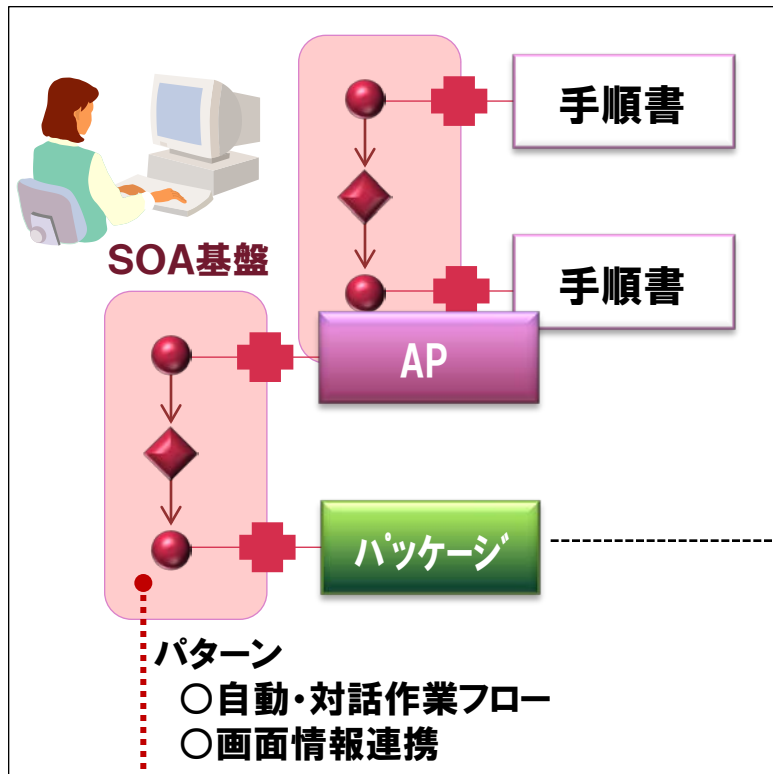
APでの作りこみ (スクラッチ開発)



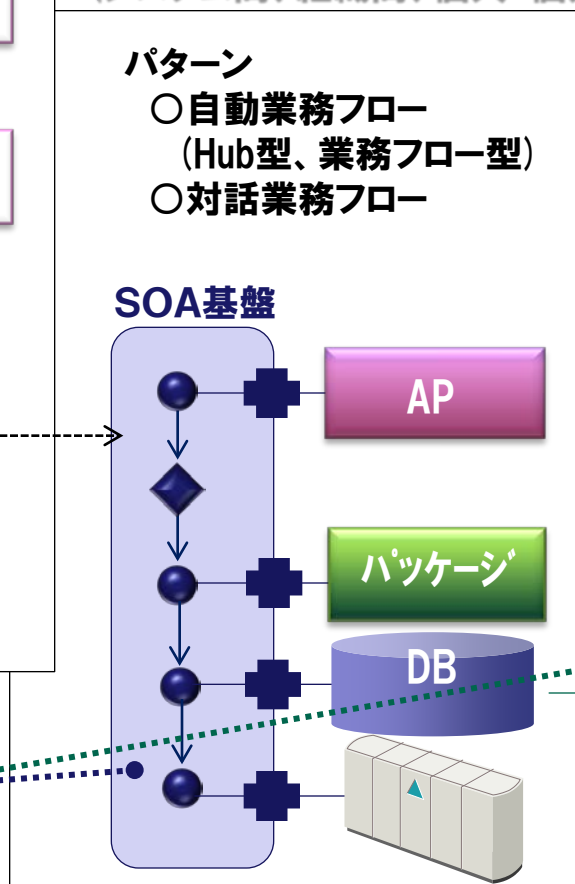
## 情報の抽出・加工処理部分

→「プロセス」部分の狭義のBAMを実現する基盤が「SOA基盤」

## 作業プロセス連携処理(個人の作業連携)



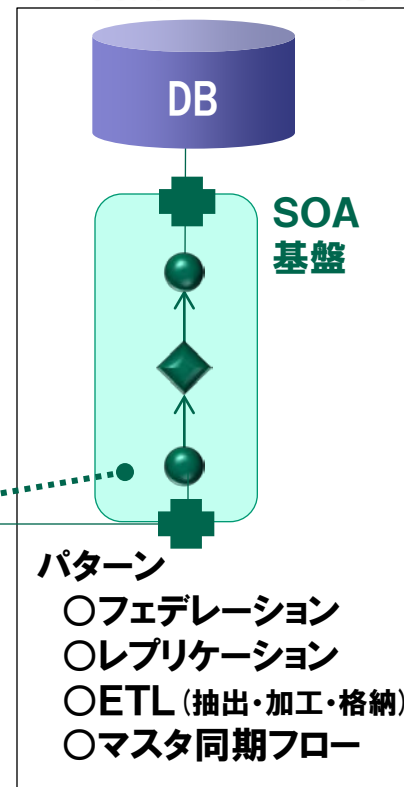
## 業務プロセス連携処理 (システム間、組織間、個人-個人)



## SOA基盤の特徴

- ・プロセス定義・実行
- ・接続アダプタ提供
- ・データ変換機能
- ・ログ出力(内部統制)

## マスタ同期プロセス連携処理



SOA基盤からのログ情報を加工して各種分析に活用

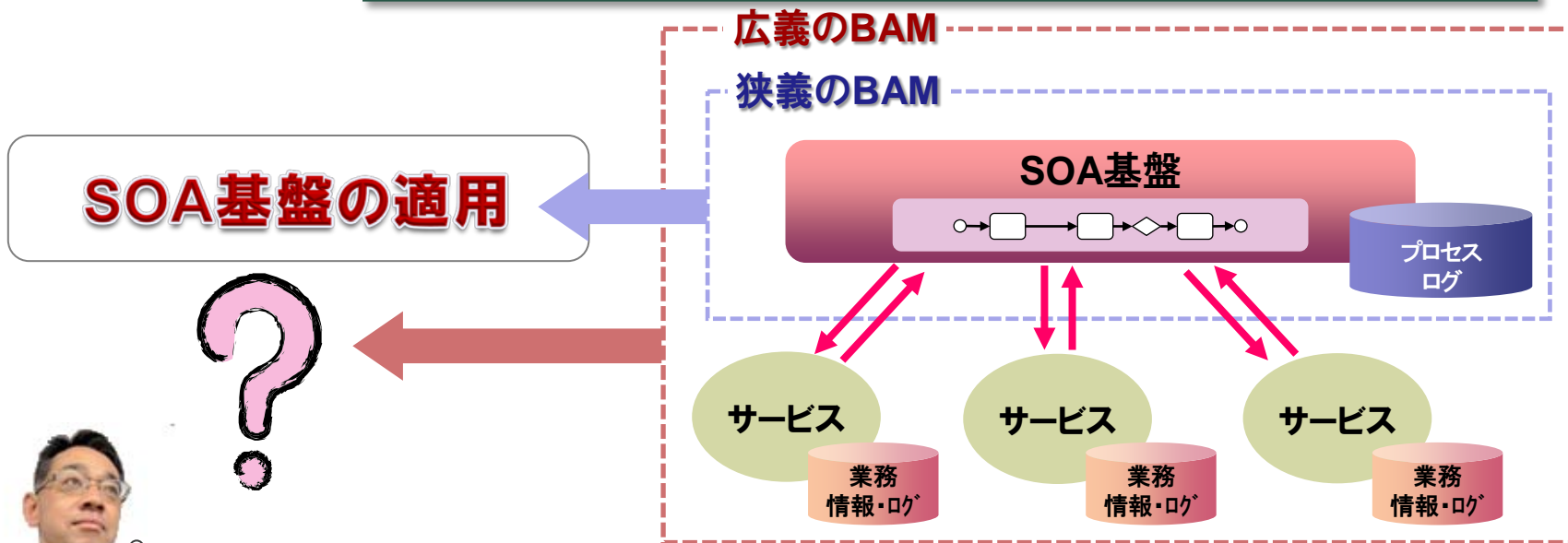
SOA基盤はプロセス制御だけでなく、経営情報の一部を支援する基盤



## BAMの課題

SOA基盤などで、**狭義のBAM**(プロセスレイヤ)は把握できても、それだけで経営情報の全てを把握したことはならない。

アプリケーションが扱う情報やログなども加味した、**広義のBAM**を実現するためにはどうすべきか？



最近、「ビッグデータ」というキーワードが巷に溢れているが、それは何者だ？

## ビッグデータの定義

一般的にはVolume、Variety、Velocityの3V（これにValueを加えるケースもある）を特性とするデータを対象に語られるケースが多い。

…が、実は、明確な定義はない !!

企業内のデータをビジネスにどう活用していくか？  
テクノロジーありきではなく、ビジネス的な視点から戦略を練るべき。

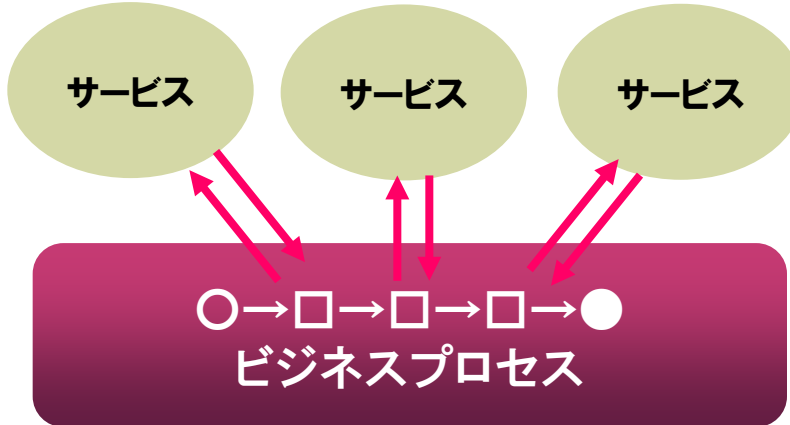
ビッグデータの  
本質は？

情報管理と情報プロセスは適切に回っているか？  
複数部門で情報を共同利用しているか？  
重複情報を抱える無駄はないか？

果たして、スモールデータなら、  
ビジネスに活用されているのか？  
改善で、付加価値を高められないか？

# 1-16 情報の受渡しプロセスのパターンは？

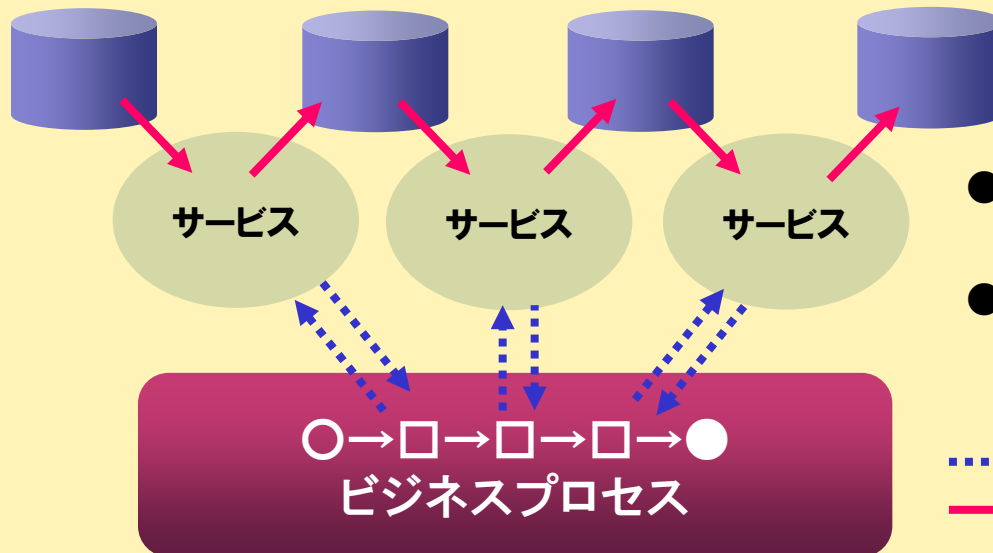
## ● 業務データ受け渡しパターン



- 制御情報と業務データを共に連携基盤が処理
- 業務データが大量になるとリソース/オーバーヘッド共に増大

- ・ EDIのデータを他社に送る
- ・ M/FとOpenでアーキテクチャが異なる
- ・ マスタが物理的な場所が分かれている

## ● 制御情報受け渡しパターン



理想系<sup>+</sup>

- 制御情報だけを連携基盤で処理し、業務データは別ルートで受け渡し
- 連携基盤は業務データを取り込まないので、リソース/オーバーヘッドは少ない

- .....▶ 制御情報
- ▶ 業務データ

# 1-17 情報管理の現状、こうなってはいないか？

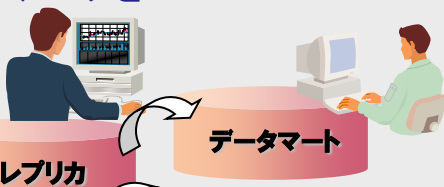
## 現状

- 業務依存型のDB設計
  - ➔ 業務間での情報の依存関係が複雑化
- システムの変化・拡大によるレプリカ、データマートなど二次情報の作成
  - ➔ 情報受け渡しの同期、情報構造変更などでの開発・運用コスト高

分析視点を変えたい



新たなシステム改修で  
データをコピー



現場の情報と  
同じものか？



書類変更に対応していない

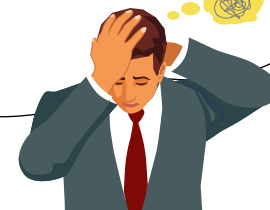


使われないのに  
生き残っているDB



指示があったが、現場では手遅れ

システムの拡大(変化)により、  
似て非なるDBや  
二次情報が増殖

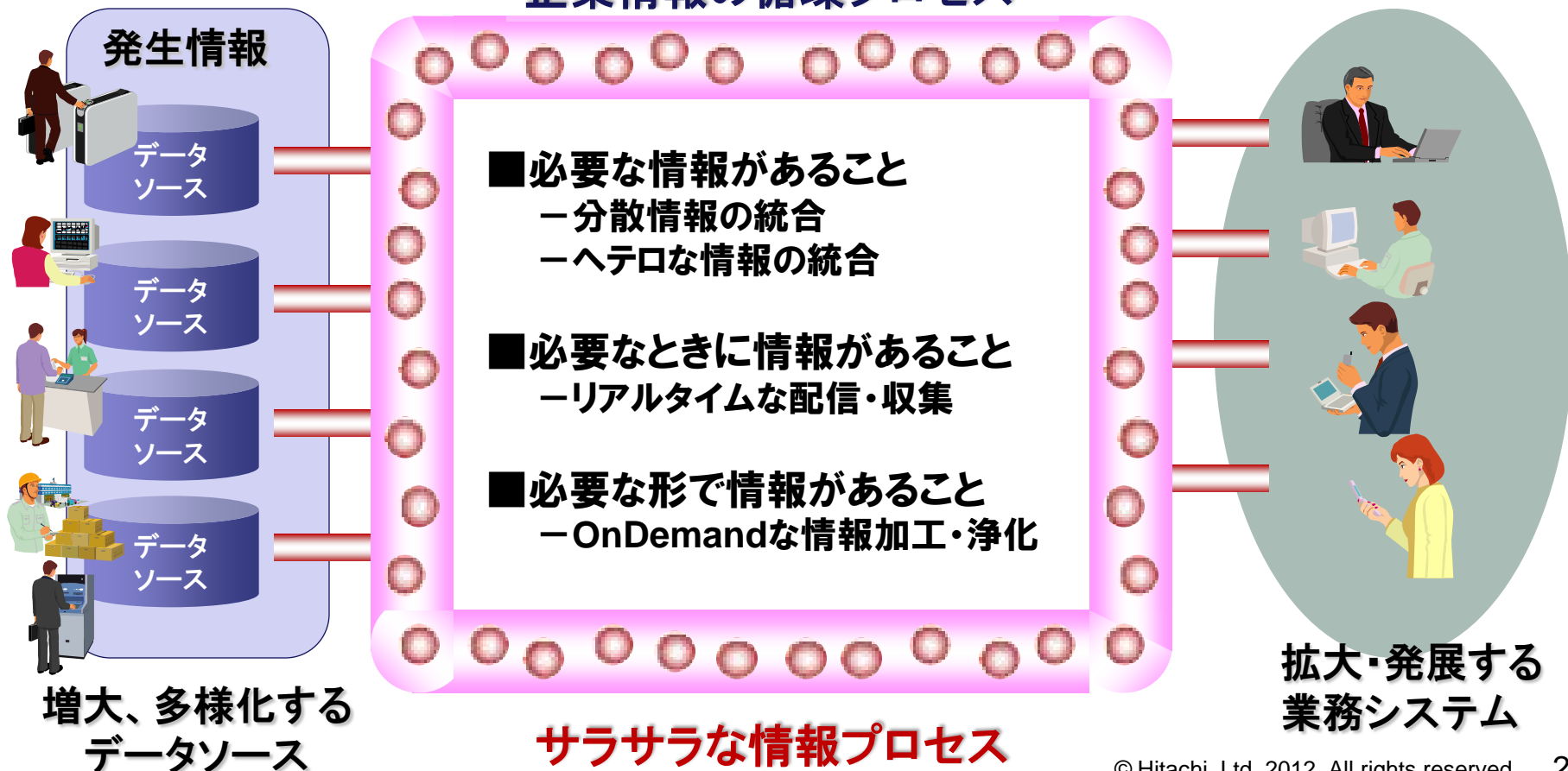


二次情報生成による  
情報プロセスの「血栓化」

あるべき姿

- 必要な情報を発生情報からオンデマンドに収集・加工・配信
- 二次情報を特定業務の固定情報としないことで、拡大・発展する業務システムへ柔軟に対応

## 企業情報の循環プロセス



**効果に着目したBPMの考え方と日立が考えるクラウド/SOA適用  
～ワークショップから得た正しい評価軸とSOA適用パターン～**

## **2. SOAの適用パターンと ワークショップから得た効果の評価軸**

## 目的

情報システム最適化に向け、日立のBPM/SOA適用に関する情報提供を行う。ケーススタディを使った参加メンバー同士のディスカッションを通じ、参加メンバーの情報システム最適化のアプローチ方法を検討する。

## 構成

- **メンバー**： ユーザ企業様10社(20名程度)  
情報システム部門において、システム企画、推進を行っている方。  
システム基盤構築や、アプリケーション開発におけるリーダ層の方。
- **コーディネーター**： 日立製作所

## 期間

約6ヶ月 期間中に4回(隔月)程度のミーティングを実施。



## 参加者メリット

- 各種情報の取得(技術動向、先進事例、SOA設計ノウハウなど)。
- 参加メンバー相互の情報交換、悩みの共有等による課題解決の参考材料の取得。
- 次期システム検討、システム企画、提案の考え方、素材の入手。





どんな点が課題？  
困っていることは？

我が社では、こんな  
工夫をしています。  
注意点は…

M&Aや追加開発で  
各システムがバラバラ。  
ノウハウ属人化で開発、  
保守コストが高い。

Cosminexusなら  
こんな風に解決できます。

使えそう！

どのくらい効果が出るの？  
参考事例が知りたい。



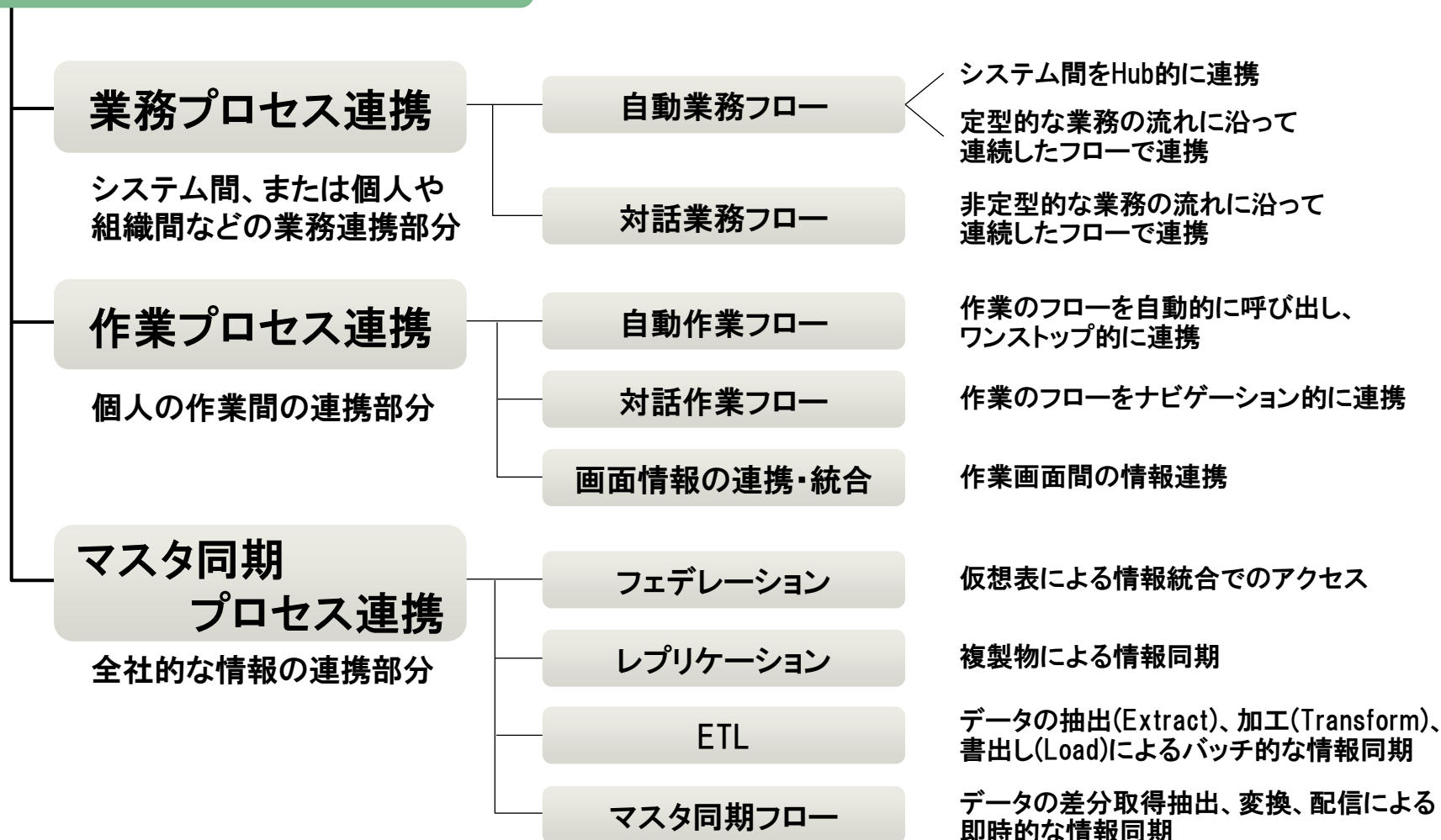


- ① コスト効果が本当に出る？いつ出る？  
出るのは、フェーズ2以降？
- ② 開発・運用効果は出たが業務主管元効果が見えない？
- ③ BPMとは業務主管元効果だけか？



**効果を整理して、疑問に答えてみよう。**

## SOAによるプロセス連携



## SOAによるプロセス連携

今回は、1つだけ  
説明します。

### 業務プロセス連携

システム間、または個人や  
組織間などの業務連携部分

#### 自動業務フロー

システム間をHub的に連携

定型的な業務の流れに沿って  
連続したフローで連携

#### 対話業務フロー

非定型的な業務の流れに沿って  
連続したフローで連携

### 作業プロセス連携

個人の作業間の連携部分

#### 自動作業フロー

作業のフローを自動的に呼び出し、  
ワンストップ的に連携

#### 対話作業フロー

作業のフローをナビゲーション的に連携

#### 画面情報の連携・統合

作業画面間の情報連携

### マスタ同期 プロセス連携

全社的な情報の連携部分

#### フェデレーション

仮想表による情報統合でのアクセス

#### レプリケーション

複製物による情報同期

#### ETL

データの抽出(Extract)、加工(Transform)、  
書出し(Load)によるバッチ的な情報同期

#### マスタ同期フロー

データの差分取得抽出、変換、配信による  
即時的な情報同期

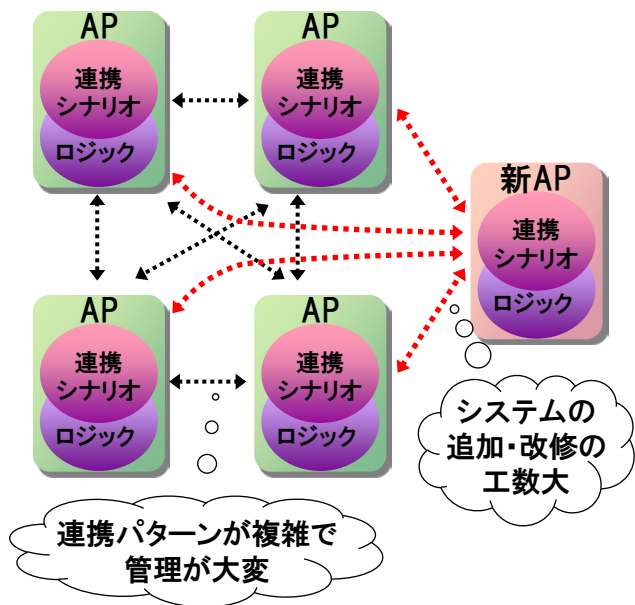
## ポイント

### 『自動業務フロー(Hub型)』による業務プロセス連携

- 複雑なシステム連携の統合・可視化によるIT運用コストの最適化と効果の早期創出
- 標準仕様に準拠した統合基盤と各種サービス接続アダプタにより、各種システムとの高い相互接続性を確保

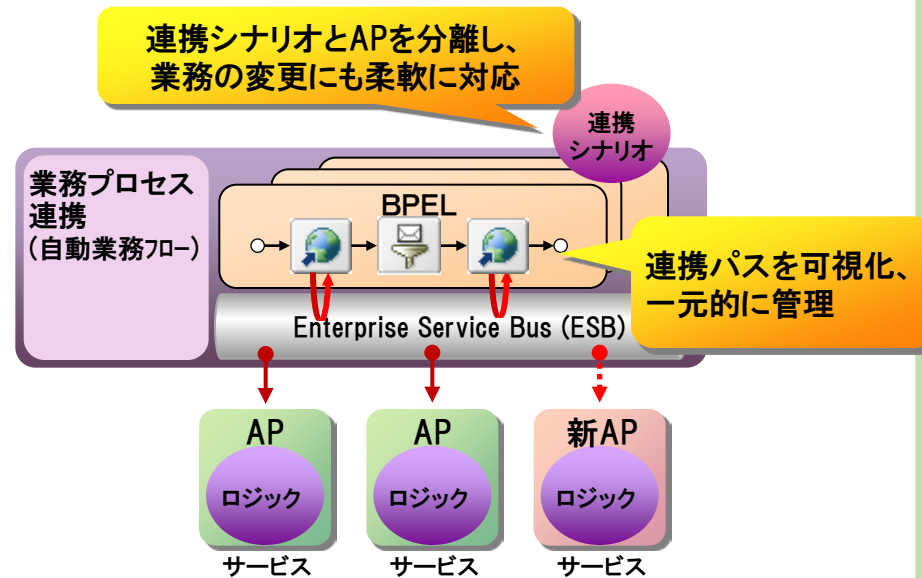
### 【適用前】

システム毎に独自インターフェースで連携



### 【適用後】

ESBをHub的に利用し、疎結合を実現



ESB/BPMIにより  
複雑なシステム  
連携を統合

# 2-7 継続的改善におけるPDCAサイクルとアクター視点

PDCAを効果的に回していくためには  
それぞれのアクターで各要素の現状  
が「**見えている**」ことが重要

ここが課題となっている  
企業が多いのが実情

業務効率向上  
業務最適化

PDCA

経営戦略

業務主管元View

流れ=プロセス

作業=AP

もの(情報)=データ

AP開発者View

運用管理者View  
(AP処理基盤に特化)

ITの視点

開発効率向上  
保守効率向上

PDCA

PDCAが回ること  
=BPMの実現

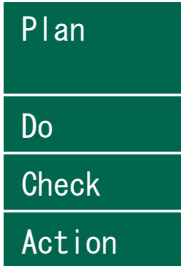
SI効率向上  
システム最適化

PDCA

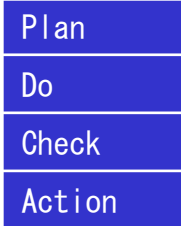
# 2-8 ITでのアクター別PDCA「見える化」のSOA部分

——→ : 自律型サイクル  
- - - - -> : 依存型サイクル

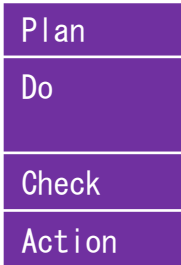
## 業務主管部門



## AP開発者



## AP運用管理者



## 流れ

業務間、作業間プロセスのシステム化計画

プロセスに沿った業務、作業の実行

プロセスの状況把握と分析

プロセスの問題点改善計画

## プロセス

プロセス設計

プロセス開発

プロセステスト

プロセス修正

## プロセス

プロセス基盤サイジング

プロセス基盤構築

プロセス基盤実行監視

プロセス稼動状況把握・分析

プロセス基盤最適化計画

## SOA基盤の導入

作業 : 作業プロセス連携  
業務 : 業務プロセス連携  
マスタ : マスタ同期プロセス連携

作業・業務・マスタ :  
・構成定義からの現状のプロセス把握  
・実行ログからの現状分析

作業 : 作業プロセス連携開発支援  
業務 : 業務プロセス連携開発支援  
マスタ : マスタ同期プロセス連携開発支援

業務 : 業務プロセス連携  
マスタ : マスタ同期プロセス連携

業務 : 業務プロセス連携基盤でリソース管理の分析用データを提供

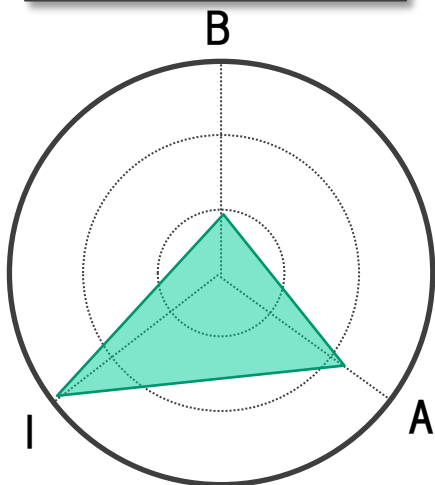
# 2-9 アクター別の基盤導入の効果

適用形態		AP開発者の効果	業務面での効果	運用面での効果
業務プロセス連携	自動業務フロー	①システム間Hub型	<ul style="list-style-type: none"> <li>④ 業務フローの自動化によるリードタイム短縮</li> <li>④ 作業効率向上</li> <li>③ 業務プロセスやプロセスログの可視化</li> <li>③ 業務分析へのフィードバック</li> </ul>	<ul style="list-style-type: none"> <li>④ 複雑な連携パスを整理して一元化</li> <li>④ APとプロセス切り離しによる運用管理の容易化</li> </ul>
		②業務フロー型		
	③対話業務フロー	<ul style="list-style-type: none"> <li>④ 新規開発の工数低減</li> <li>④ 保守時の調査工数低減</li> <li>③ プロセス部分保守時の改修工数低減</li> </ul>		
作業プロセス連携	④自動作業フロー	<ul style="list-style-type: none"> <li>④ 連携パスの一元化による改修時の調査工数低減</li> <li>③ 連携パスの一元化による改修時の改修工数低減</li> <li>③ APとプロセスを切り離し、条件変更時のAP保守工数を低減</li> </ul>	<ul style="list-style-type: none"> <li>④ 作業ステップ間の連携により、リードタイムを短縮</li> <li>④ 作業のシステム化により、作業順序・内容を平準化</li> <li>③ 作業プロセスやプロセスログの可視化</li> <li>③ 作業分析へのフィードバック</li> </ul>	<ul style="list-style-type: none"> <li>④ プロセスと連携システムとの切り離しによる運用管理の容易化</li> <li>③ 作業プロセス変更の容易化</li> </ul>
	⑤対話作業フロー			
	⑥画面情報連携	<ul style="list-style-type: none"> <li>④ 画面の統合による使いやすさ向上</li> <li>④ データ連携による作業効率向上・ミスの低減</li> </ul>		
⑦マスタ同期プロセス連携	フェデレーション	④ 移行時の開発工数低減	<ul style="list-style-type: none"> <li>④ マスタ情報の一元管理を実現</li> <li>③ 新規データ追加時などの業務で早期利用が可能</li> </ul>	④ 情報の一元化による管理コスト削減
	レプリケーション	④ マスタ同期時の開発工数低減		
	ETL			
	マスタ同期フロー			③ 連携プロセスの変更の容易化

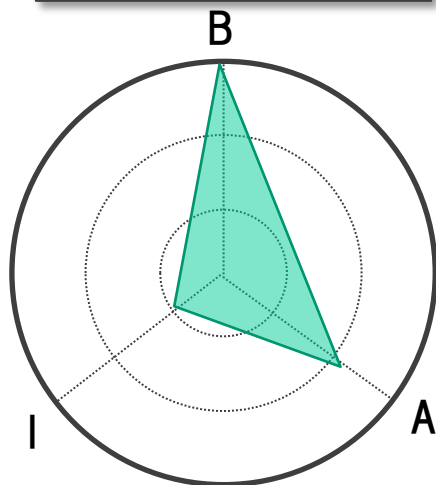


# 2-10 アクター別効果をチャートで表わすと

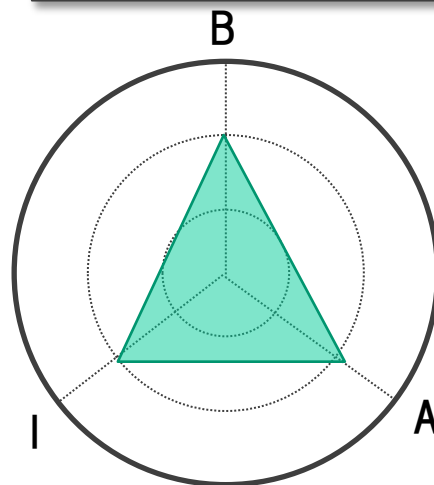
①業務プロセス  
(Hub型)



②業務プロセス  
(自動フロー)

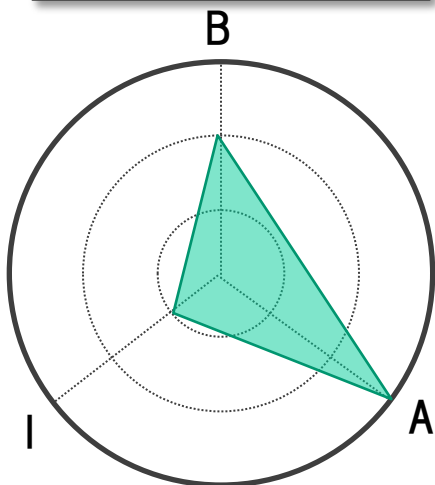


③業務プロセス  
(対話型)

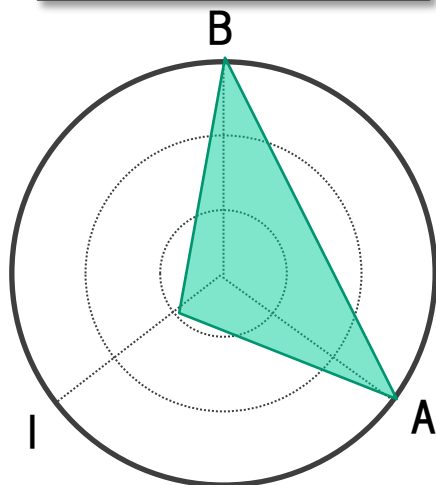


B: 業務主管部門  
A: AP開発者  
I: 運用管理者

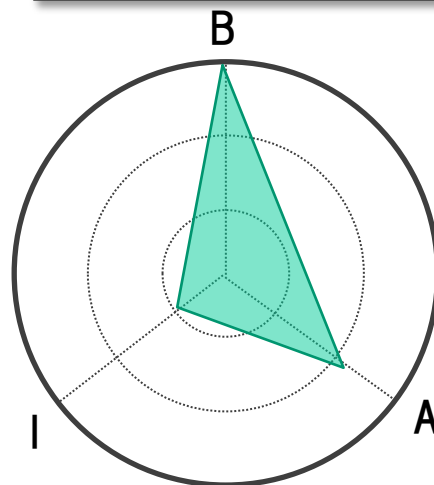
④作業プロセス  
(自動作業フロー)



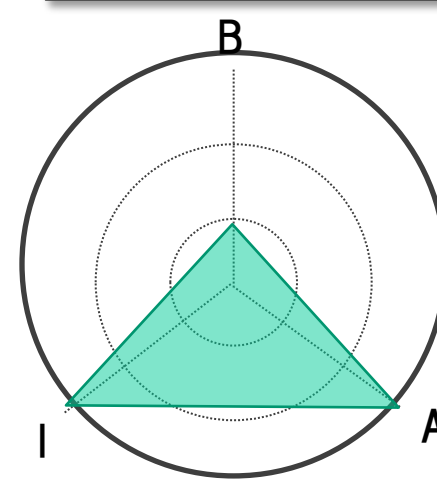
⑤作業プロセス  
(対話型)



⑥作業プロセス  
(画面間連携)



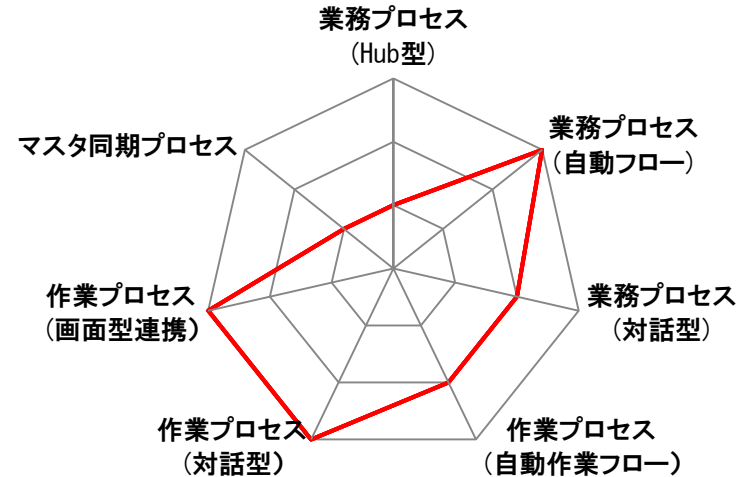
⑦マスタ同期プロセス



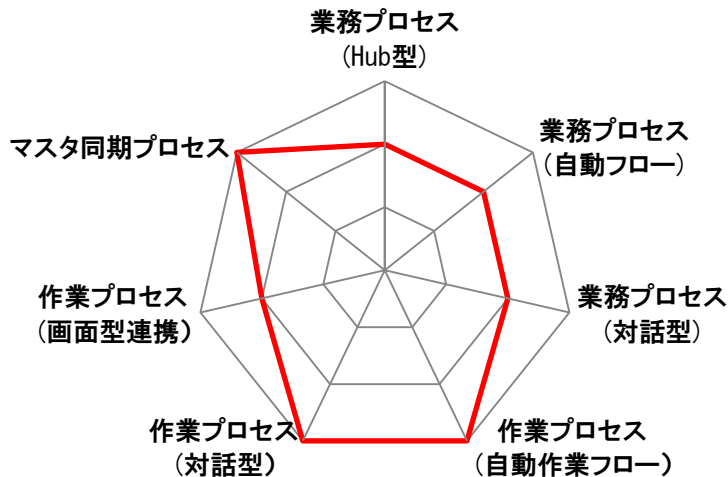


- これらはいくまで導入の優先を決めるためのひとつの指標である
- 企業において、現状の課題を明確にし、その上でこれらの効果を見据え、どれから整備するかは企業により異なるはず
- ただし、「BPM」=「各アクターのPDCAを回すこと」が重要であり、これらは経営戦略に直結する
- システムを構築することだけに留まらず、企業のPDCAを回すことが**マネージメント**であるはず

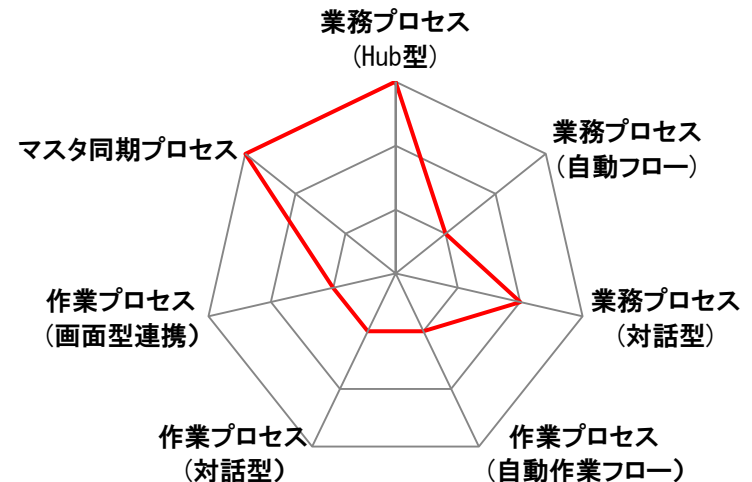
## 業務主管部門



## AP開発者



## 運用管理者



## 業務要件

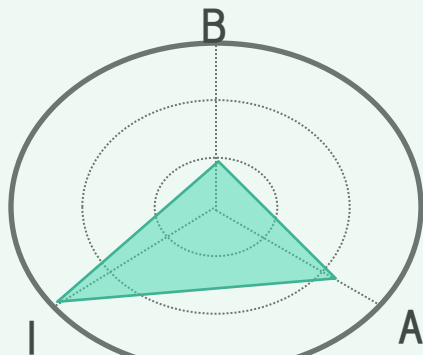
決算・取引先・人員情報の  
月末処理のスピードアップ

## 情報システム部の課題

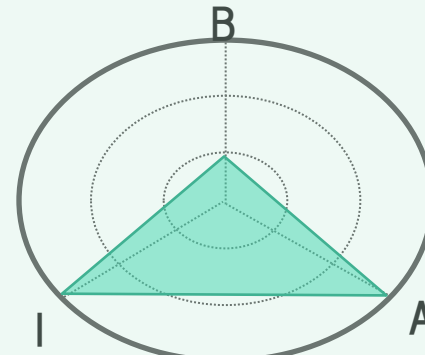
開発コストの大幅削減

2つのパターンでアプローチ

### ①業務プロセス(Hub型)



### ⑦マスタ同期プロセス



### システム間連携の現状…

- ▶ 今日までシステム間連携の標準方式無し
- ▶ コスト最小化のため安易にFile連携を採用
- ▶ 最終的にFile連携が増加し、保守運用コストが増大
- ▶ バッチ処理的なFile連携により、リアルタイム連携へ壁

誰もが簡単かつ標準のI/Fで連携できる基盤が必要

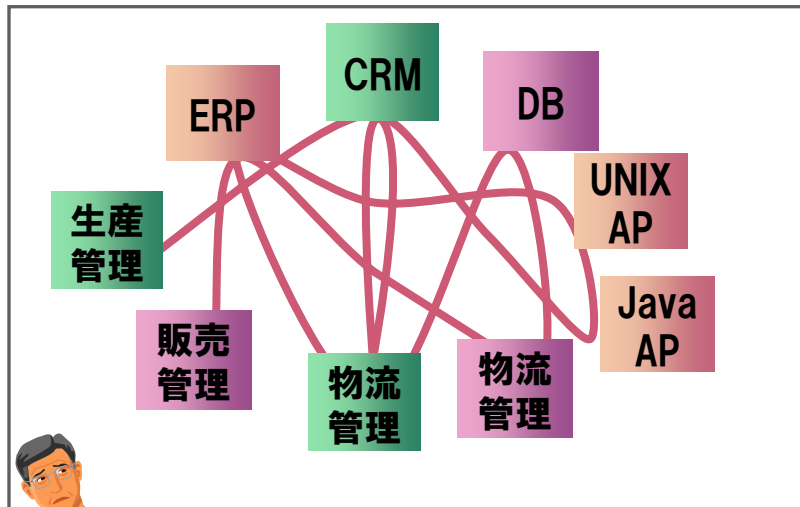
SOA基盤によるシステム間連携の標準化と展開を推進

SOA基盤  
将来展望

- ・1つの巨大な情報連携／サービス公開基盤として運用
- ・各社システムが持つサービスや情報をSOA基盤により連携させ、サービス展開・活用を拡大

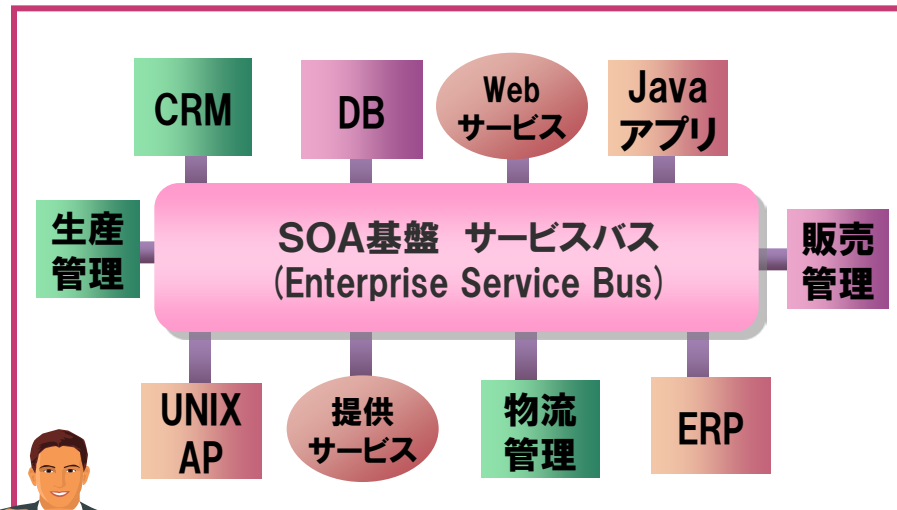
## シームレスな情報連携を実現する基盤です

### 個別のシステム同士が接続する形態



- 個別システム同士の連携が増大  
⇒各機能の再利用・共有化が難
- ファイル連携などバッチ的連携が主  
⇒リアルタイム化へ壁
- 連携はスクラッチ開発、I/Fはバラバラ  
⇒連携が増えるたびにコスト増大

### SOA基盤を適用後



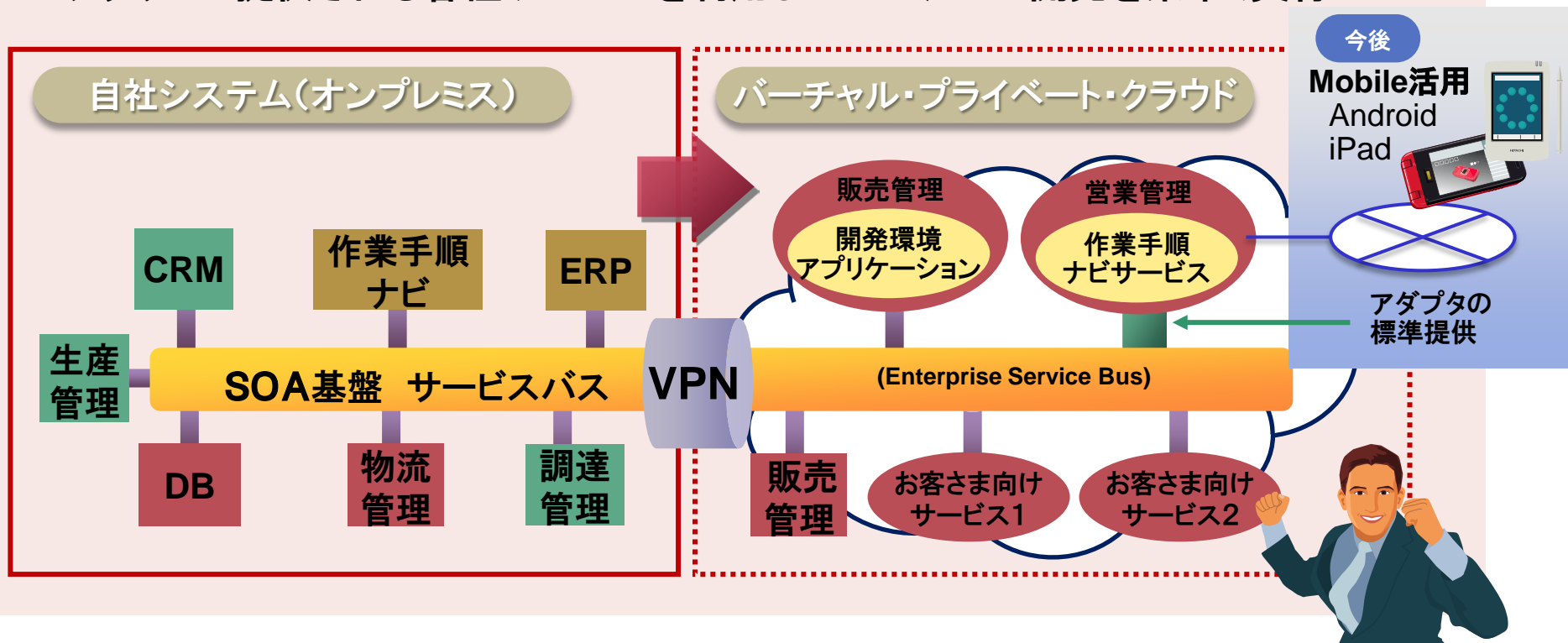
### SOA基盤の活用メリット

- 開発コスト : 65%削減  
⇒DBからの抽出PG、格納PG開発不要
- 運用コスト : 50%削減  
⇒SOA基盤で集中監視。システム運用負担減
- インフラ導入コスト : ゼロ  
⇒連携転送に関するサーバ、ミドルソフト不要

## 企業の成長を支え続けます

### これからの情報システム

- SOA基盤でシステムを整理/整頓して拡張性を確保
- バーチャル・プライベート・クラウドで、自社システムをセキュアに拡張、オンプレミスとクラウドで適所にシステムを配置してオフバランス
- クラウドで提供される各種サービスを利用してシステムの開発を素早く実行



**効果に着目したBPMの考え方と日立が考えるクラウド/ SOA適用  
～ワークショップから得た正しい評価軸とSOA適用パターン～**

### **3. 「見えるクラウド」を実現するための取り組み**

# 3-1 これからの企業に求められるITの役割

これまで

これから

経営層の要求

先進的で、良いシステム、役に立つシステムを構築して欲しい。

最小コストで品質と生産性の高いITシステムを実現して欲しい。

➡ クラウドを上手に使うって効果を出せないか？

業務部門の目標

- 事業単位でシステム化推進。
- 業務拡大と効率化に注力。

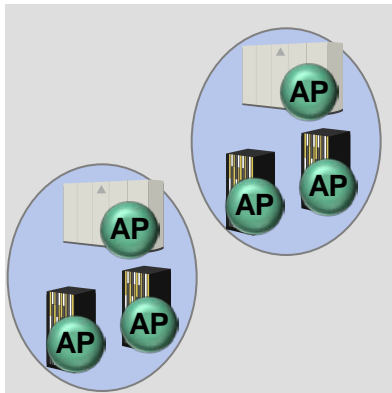
- 事業の選択と集中。
- グループ統廃合、M&Aによる業務の集約化。

- 業務連携(部署連携)、スループットの向上。
- 新商品、新業務のスピード立ち上げ。

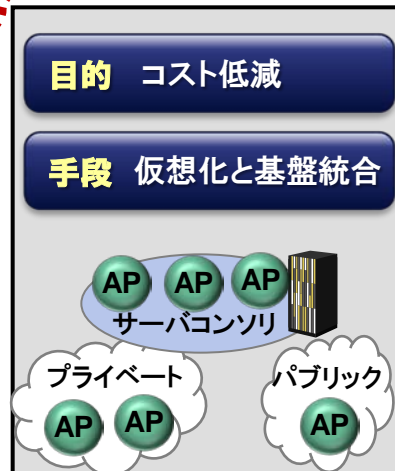
- 多様なサービスと連携。
- 新分野への挑戦。

IT部門の役割

オンプレミス

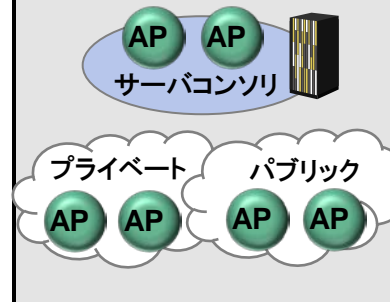


フェーズ 1



目的 スピード開発

手段 クラウドの活用

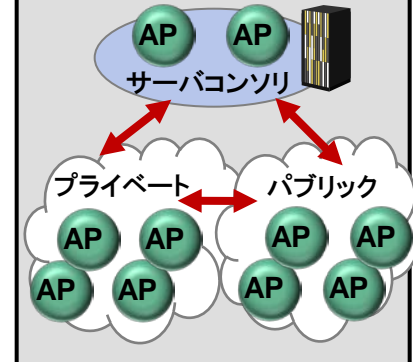


フェーズ 2

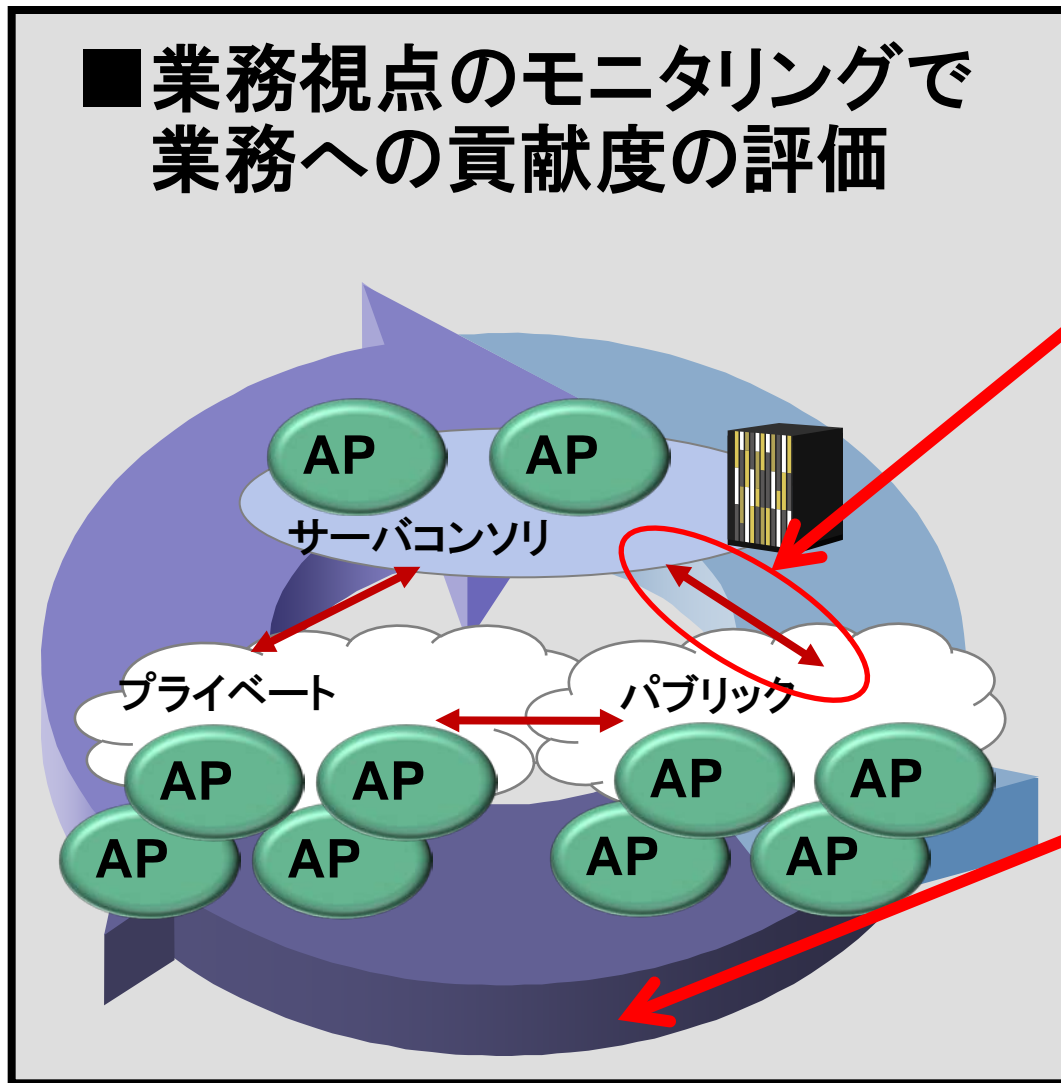
フェーズ 3

目的 新ビジネス創出

手段 サービスの連携



## ■業務視点のモニタリングで 業務への貢献度の評価



### サービスの連携(つなぐ)

- ・つなぐものが増えた。
- ・つなぎ先が増えた。

### ■SOAに基づいて連携

- ・標準技術での連携
- ・疎結合

### プロセスの評価(見える)

- ・クラウドは中身が見えない
- ・プロセスのブラックボックス化

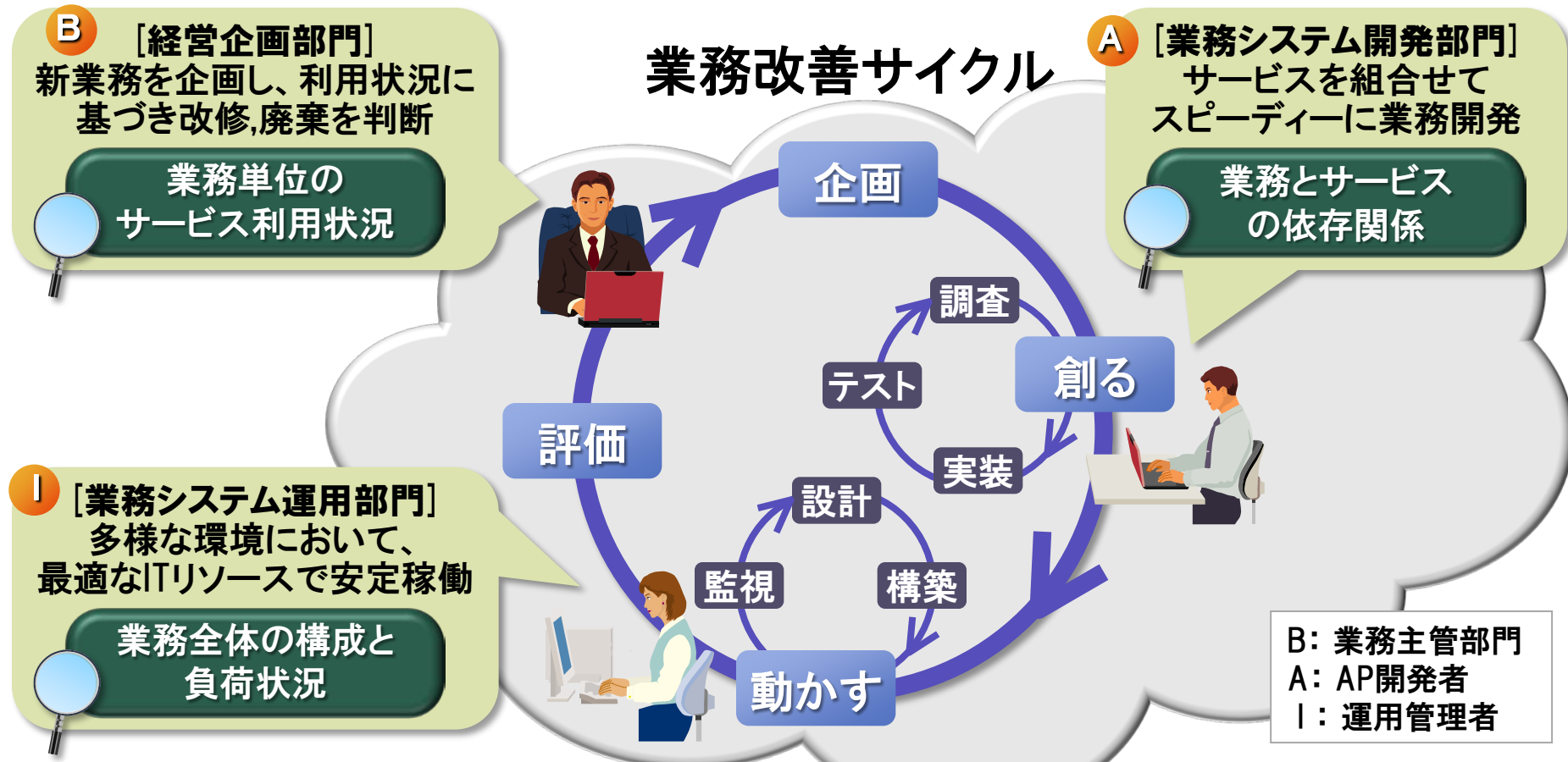
### ■BPMに基づいて見える化

- ・稼働状況のモニタリング
- ・業務プロセス評価・検証



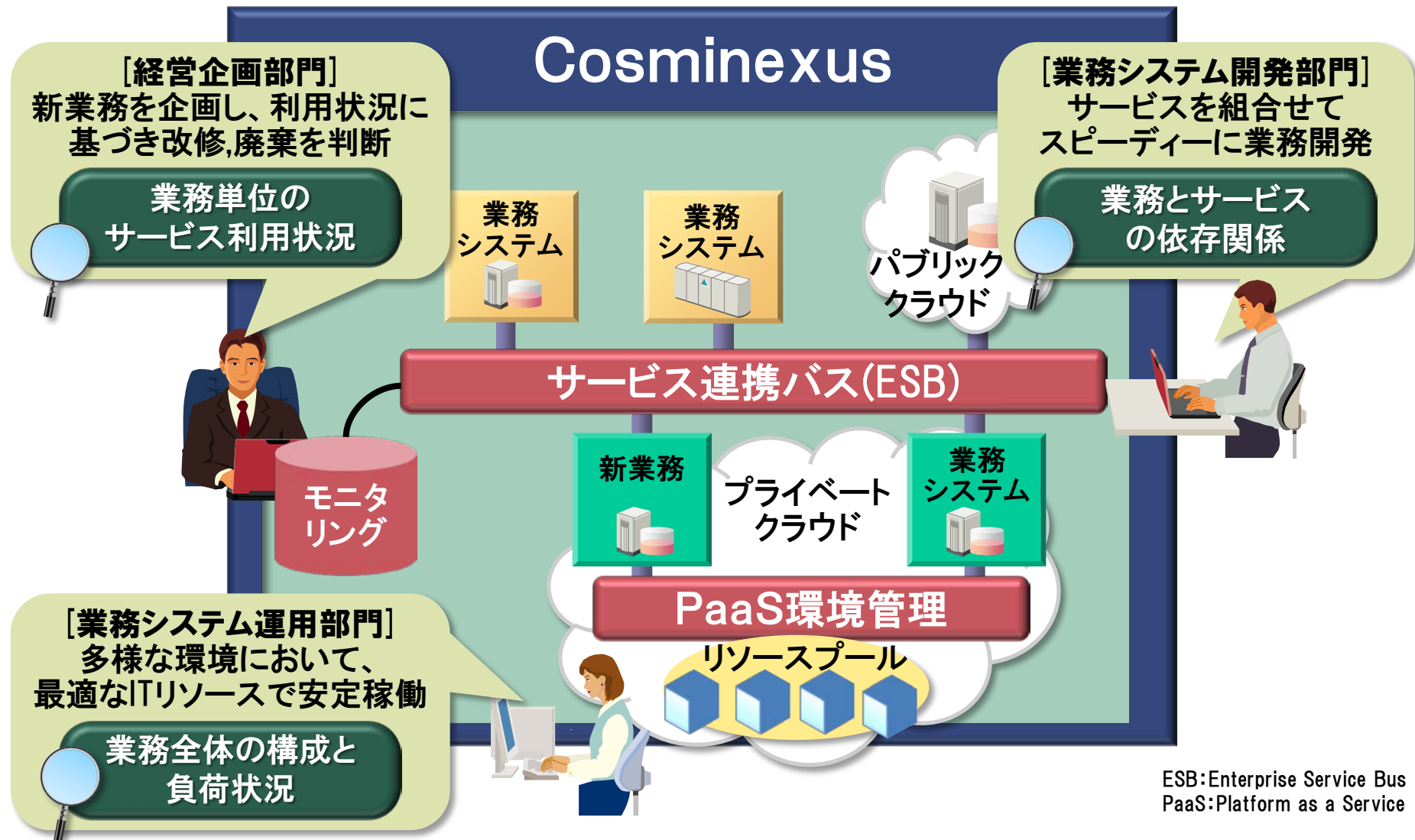
## ポイント

- 業務視点で可視化する仕組みを提供し、業務改善サイクルを加速



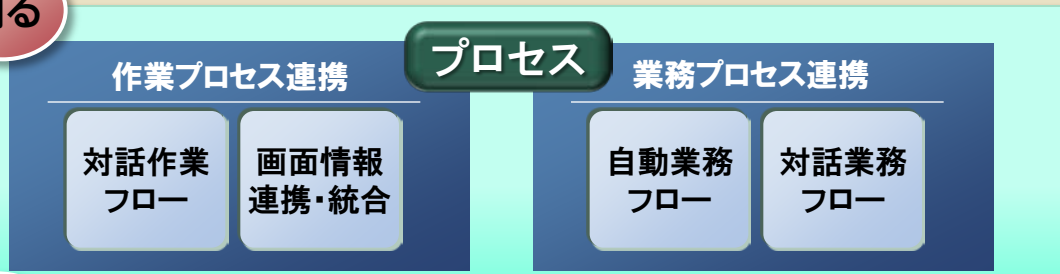
日立が考える「見えるクラウド」

# 3-4 見えるクラウドを実現する Cosminexus

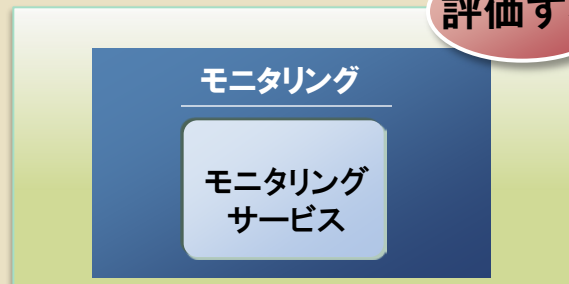


## クラウドサービスプラットフォーム *Cosminexus*

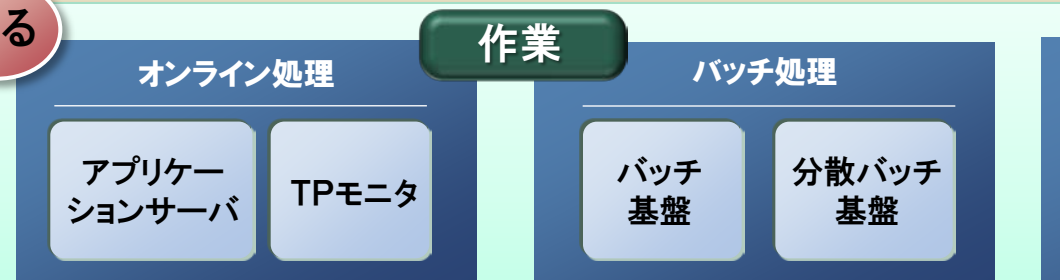
創る



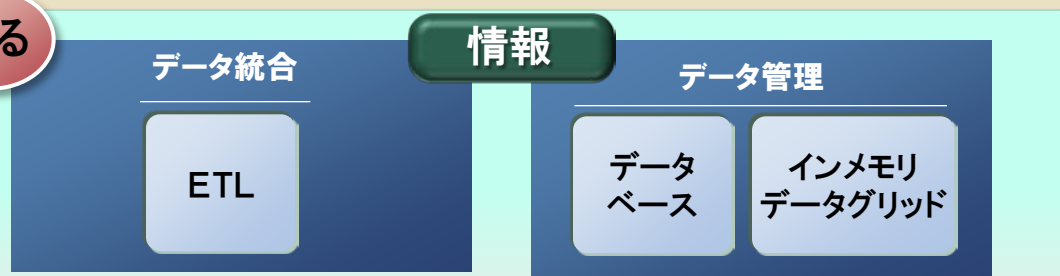
評価する



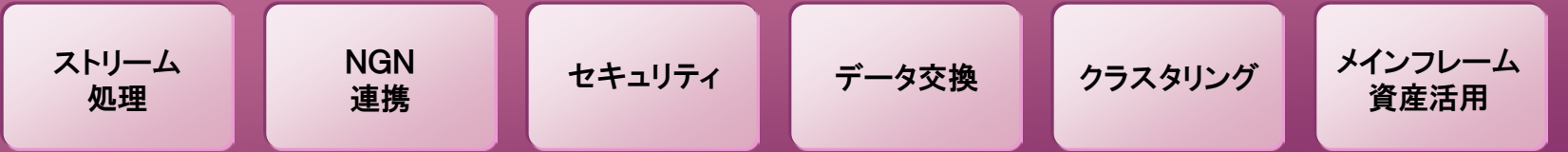
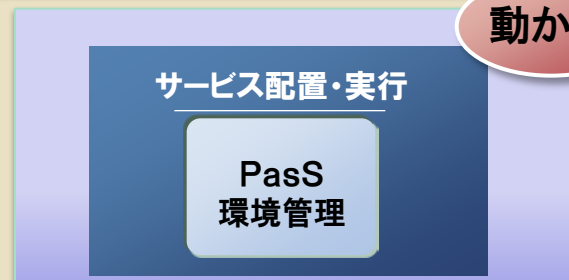
創る



創る



動かす



**効果に着目したBPMの考え方と日立が考えるクラウド/SOA適用  
～ワークショップから得た正しい評価軸とSOA適用パターン～**

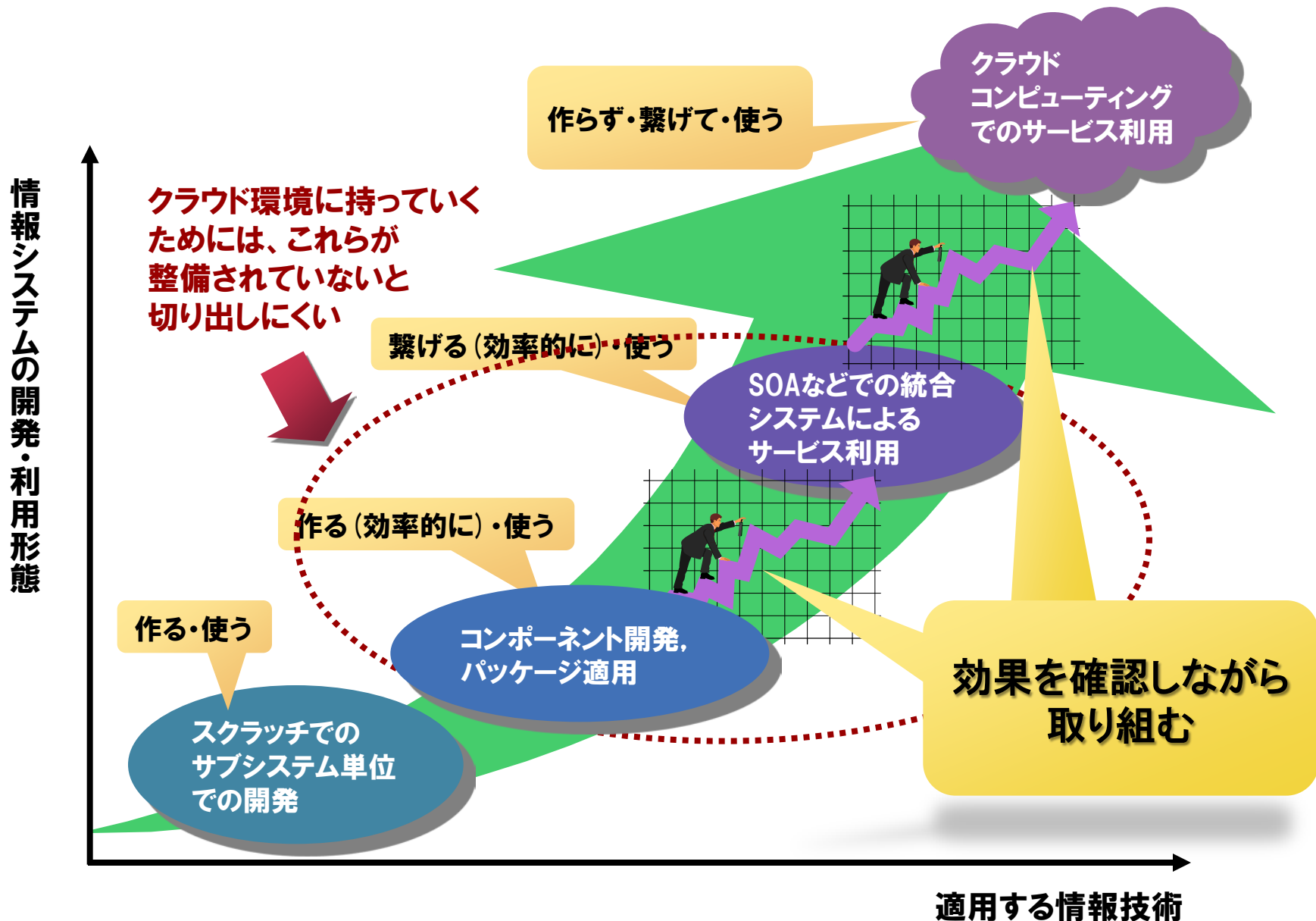
## **4. まとめ**

# BPMを実現するSOA適用の秘訣

- ① アクター別の効果にそって適用モデルを選択
- ② ミドルウェア活用ソリューションで  
適用イメージフィージビリティを検証してスタート
- ③ SOA化は、クラウド化および将来のビッグデータ  
活用基盤にも繋がる！

**効果に着目したBPMの考え方と日立が考えるクラウド/SOA適用  
～ワークショップから得た正しい評価軸とSOA適用パターン～**

**付録**



効果や実現性を確認しながら進めたい



提案段階から技術検証を支援するメニューと体制を整備します。  
「ミドルウェア活用ソリューション」

企画

計画

要件定義

設計

構築

テスト

本番

企画サービス

顧客要件から  
次期システム像を  
ご提案

スタータサービス

実現性、開発工数算  
出、性能検証などの事  
前評価を実施

テクニカルサービス

設計・開発に関する問い合わせ、チューニング、設計、  
性能測定、テストなどを支援





## ポイント

- 次期システム像(オリジナルモデル)をご提案するサービスです。
- 課題解決をモデル化したユースケースを準備しています。

初期プレ

お客様の  
モヤっとした要件

提案概要検討

RFI

回答

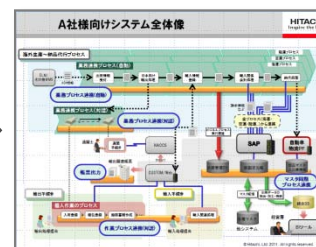
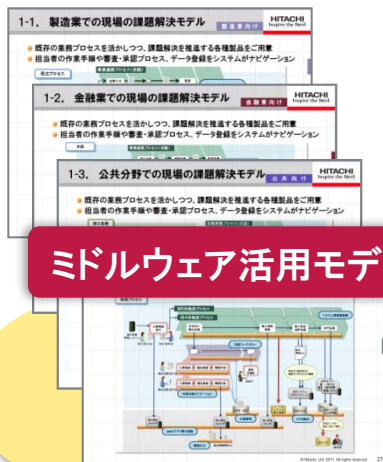
提案書作成

RFP

提案

RFI: Request for Information

RFP: Request for Proposal



オリジナルモデル

ソリューション  
適用提案

ミドルウェア  
適用提案

「モデル」や「ユースケース」を利用して、モヤっとした要件を整理し、お客様のオリジナルモデルを作成します。



## ポイント

- RFP前に、適用効果・実現性の検証を支援するサービスです。
- 各種アセスメント、プロトタイプ作成、教育などを支援します。

初期プレ

お客様の  
モヤっとした要件

提案概要検討

▼ RFI

回答

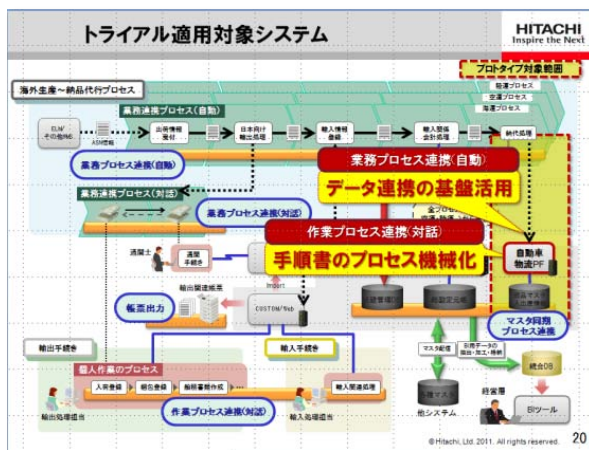
提案書作成

▼ RFP

提案

RFI: Request for Information

RFP: Request for Proposal



トライアル適用範囲を決定

仕様書

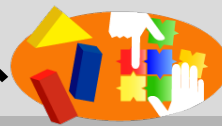
トライアル実施

検証

具体的なサービス内容

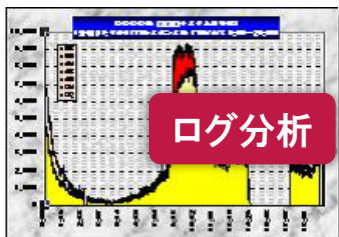
- 業務におけるミドルウェア適用の  
アセスメントを実施します。
- 評価のためのプロトタイプ作成や  
技術支援を実施します。
- ミドルウェア適用の手順を  
提示します。

実現性、開発工数算出、性能検証などの事前評価を実施することで、  
精度の高いシステム計画をサポートします。



## ポイント

- システムの設計開発・安定稼働を支えるサービスです。
- 設計・開発に関する問い合わせ、チューニング、設計、性能測定、テストなどを支援します。
- 既存の「テクニカルサービス」と同様の内容です。



### 具体的なサービス内容

- 設計書や手順書を専門の技術者が確認します。
- 環境構築やテストに専門の技術者が立ち会います。
- 設計・開発のテクニカルスキルを教育します。



## 《他社所有名称に対する表示》

- Microsoft .NET は、お客様、情報、システムおよびデバイスを繋ぐソフトウェアです。
- UNIXは、The Open Groupの米国ならびに他の国における登録商標です。

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。

**END**

---

**効果に着目したBPMの考え方と日立が考えるクラウド/SOA適用  
～ワークショップから得た正しい評価軸とSOA適用パターン～**

2012/11/9

株式会社 日立製作所 情報・通信システム社  
ソフトウェア本部 システム基盤ソリューション部

**吉村 誠**