



連続セミナー2006
第1回「CIOを取り巻く環境」

情報技術の変遷

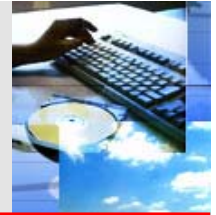
システム構築・運用におけるソフトウェア技術に焦点をあてて

2006年6月9日

上原三八

富士通研究所

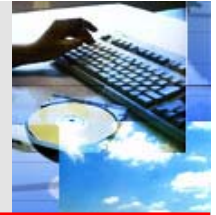
内容



- はじめに
- 分散ミドルウェア
- 仮想と自律
- システム・セキュリティ
- ビジネスの為のソフトウェア工学
- おわりに

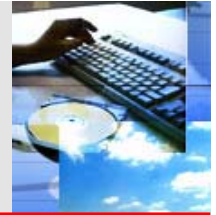
入門編では
ありません

システム構築の変化



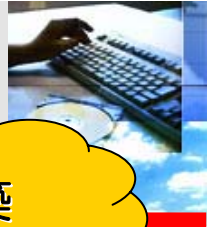
- 従来の事務作業の効率化から、ビジネス・オペレーション基盤としてのシステム作りにIT投資がシフト
 - システム作りの成否が企業の業績を左右
- 既存システムを含む、複数のシステムを連携させることが増えつつある
- インターネットに始まる技術革新により、インフラやミドルウェアの利用技術はますます複雑となり、スキル習得や蓄積が重要な課題となっている

ソフトウェア技術は着実に進展



- 技術は着実に進展している
 - 《実態》益々巨大で複雑なシステムを構築
 - 《誤解しないで》「理想やアイデアは昔からあった」と、実現できていることは別！
 - 「対象範囲を絞ればソフトウェアはできてしまう」ことがソフトウェアの認識を誤らせるもと
- しかし、ベンダー間の主導権争いが技術の進展・普及を阻害
 - ベンダー間の接続性/整合性が益々大事に

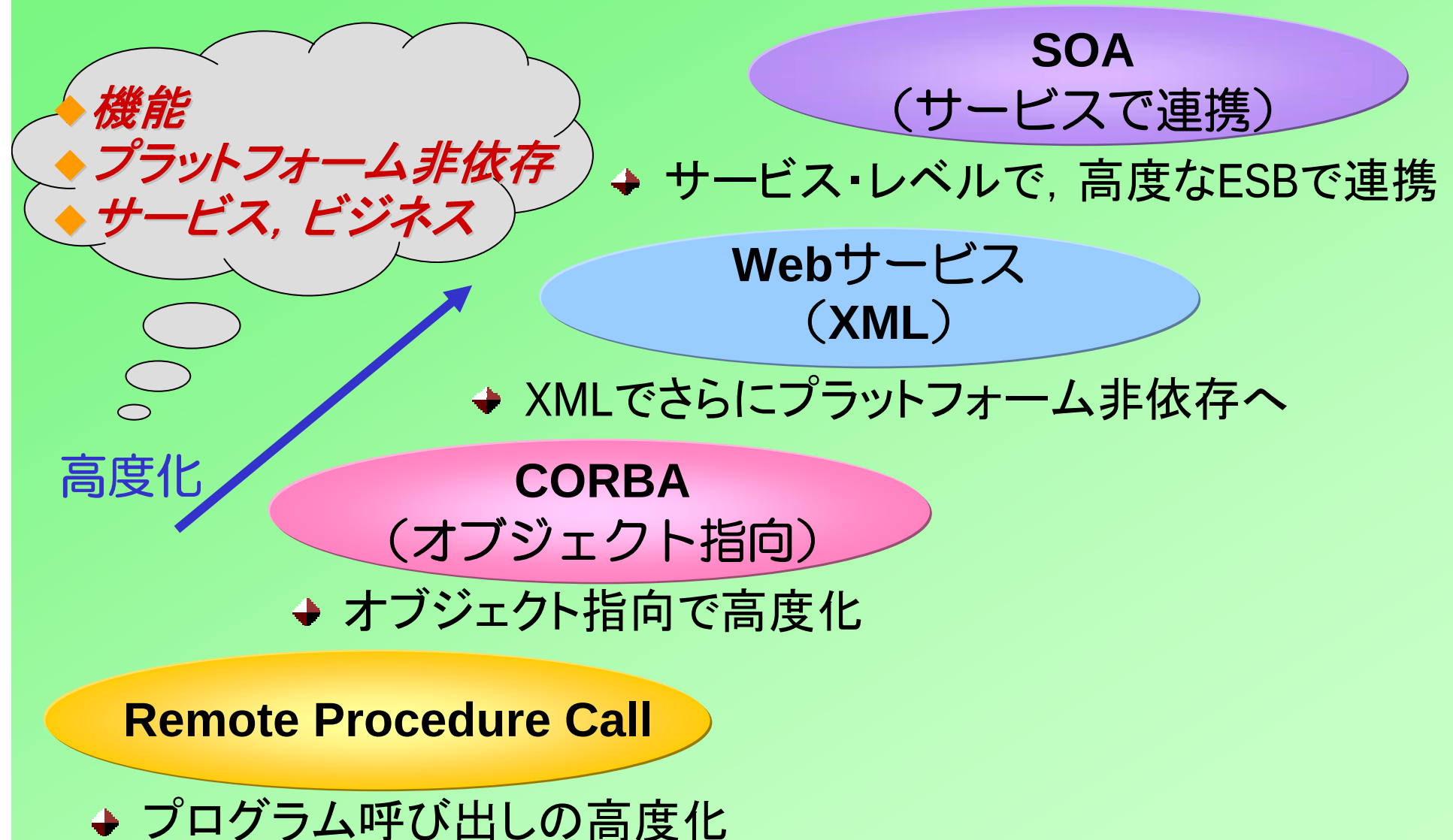
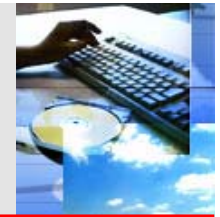
1. 分散ミドルウェア



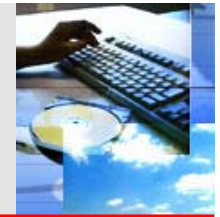
システム構築
の観点から

- 分散ミドルウェアの定義:「ネットワーク上でサービスの依頼者(client)と提供者(server)を結びつけるソフトウェア」
- 分散ミドルウェアの発展は、まさに複雑なシステムを組み上げるための要(かなめ)
 - *Remote procedure call*から, *Web services*そして *SOA/ESB*へ

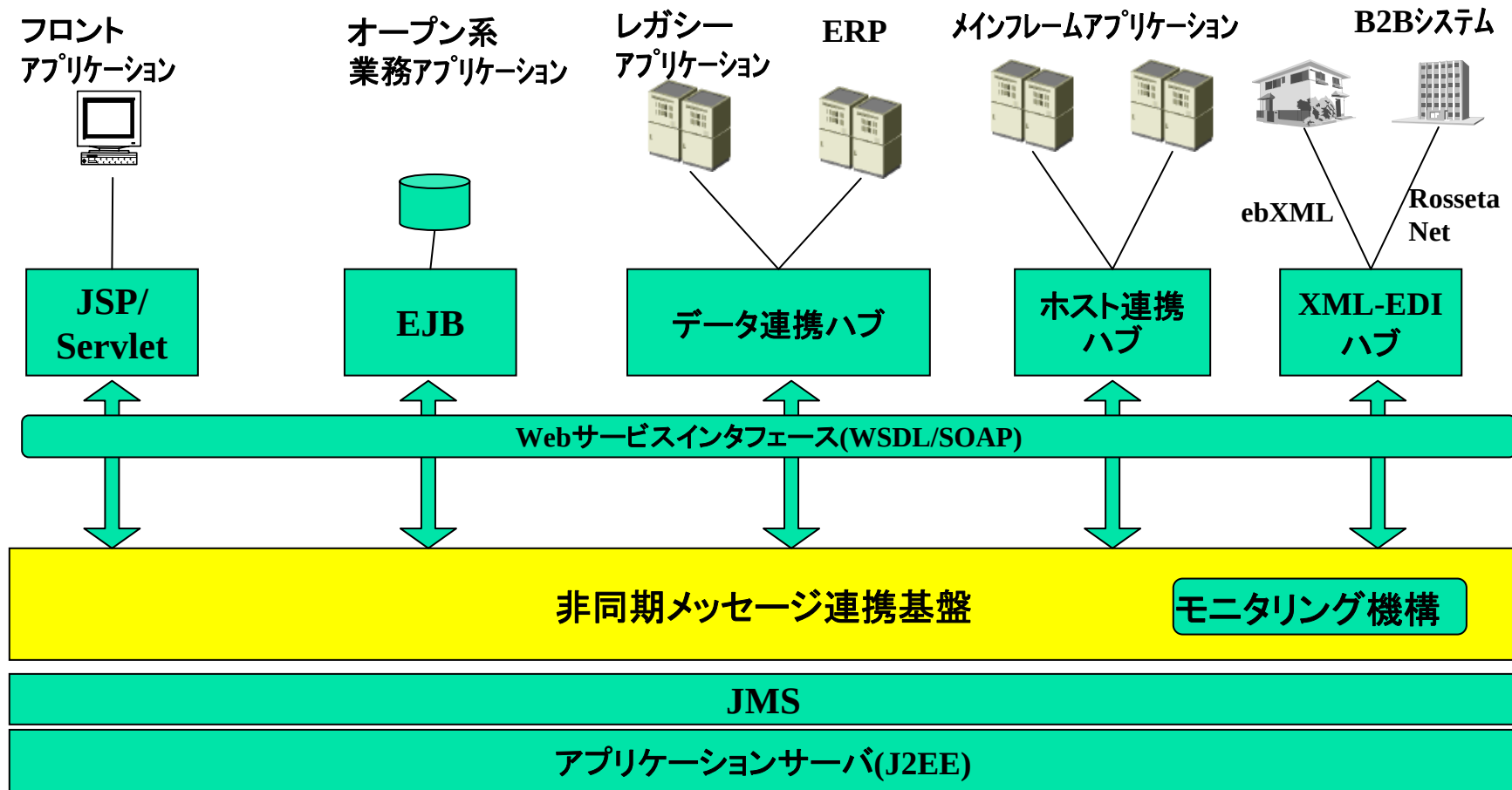
分散ミドルウェアの発展



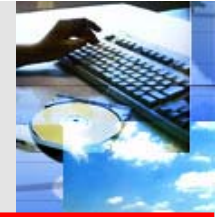
メッセージ連携アーキテクチャの例



- オープン系、レガシー系のアプリケーションをXMLベースのメッセージングにより連携



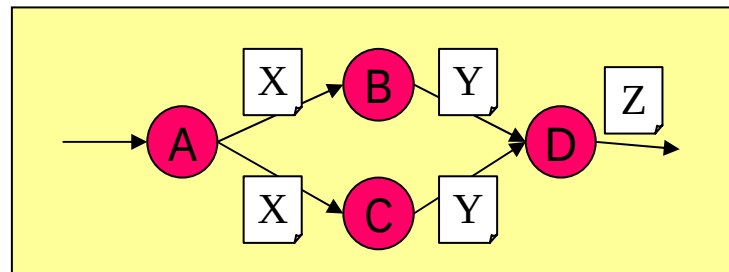
疎結合で柔軟なシステムの重要性



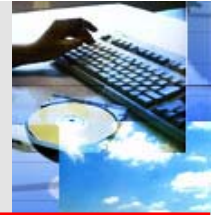
- 業務の変更や追加に対して、既存部分の変更が少ないシステムが重要。特に、サブシステムとその連携部分の開発を分けて、変更や拡張の独立性を高めることを目標とすべき。
 - 業務システムの追加や業務フローの変更の際にリコンパイルが不要
- 制御サーバが各システムと独立して障害に対して強いこと
 - ある業務システムが立ち上がっていない、業務システムやネットワークに障害が起きた際でも、他業務システムの実行に影響を与えない
- 開発面だけでなく、テスト範囲を極小化できることが重要
 - 一般的には、処理フローよりむしろデータフローの観点から疎結合とすれば、システム変更時のテスト範囲を狭めることができる。

業務プロセス

A～D: サブシステム
X～Z: データ

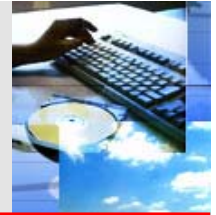


疎結合システムの開発法の必要性



- 従来のアプリケーション開発のスコープは、個々の要件を満たすシステムを作り上げることであった。しかし、昨今の業務システムでは、迅速なvalue chainの構築・変更を目的とするシステム連携が必要となり、しかもそれは事前に予測できない。そこで、業務拡張にも強く、かつ迅速なシステム連携が可能なシステム開発法が必要となってきた。例えばPublish-subscribeのパターンを基本とした、システムを疎に連携させる設計法がある。
- 従来の分散通信ミドルウェア(Webサービス, CORBA)はネットワーク上のサブシステムを透過にする(RPCする)為のメカニズムであり、それだけでは疎なシステム構築が可能となる訳でない。
※SOAPなどのWebサービスの技術は、XML文書をメッセージ交換させることで、CORBAよりプラットフォームおよびプログラミング言語からの独立性を高めている。
- ワークフロー・エンジンによる分散システム統合は、重要な技術であるが、それだけでは十分でない。

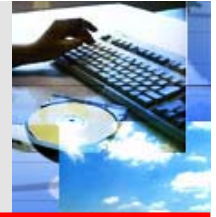
例) アプリケーション連携パターン



- Pipes-And-Filters
 - パイプ処理の中身は隠蔽される
- Competing Consumers
 - ただひとつのコンシューマがメッセージを処理できる
- Context-Based Router
 - 外部のロジックによって振り分ける。バックアップ用など
- Content-Based Router
 - メッセージの中身によって振り分ける
- Publish-Subscribe Channel
- Recipient List
- Aggregator

参考: Enterprise Integration Pattern (Addison-Wesley)

例) Recipient List vs Pub-Sub/Filter



- 発行者が受取人をコントロール
 - 集中管理
 - ルータが受取人を知っている
 - 予測的ルーティング
- × 発行者が受取人を意識しなければならない

- メッセージをコピーして振り分けたあとフィルタリング
 - 分散管理
 - 受取人を知らなくて良い
 - 受動的フィルタリング
(オープンオークション)
- × 効率の良い実装に工夫が必要

2. 仮想と自律

システム運用
の観点から

ユーザ要求

■ 迅速 Agility

- Speedy adaptation to changing business needs
- Quick response to new opportunities
- Fast development of new systems

■ 効率 Efficiency

- Lower TCO
- Higher utilization of existing resources
- Reduced operation costs

■ 事業継続 Continuity

- 24 hours a day, 7 days a week
- Dynamic configuration
- Security

IT技術

■ 仮想 Virtualization

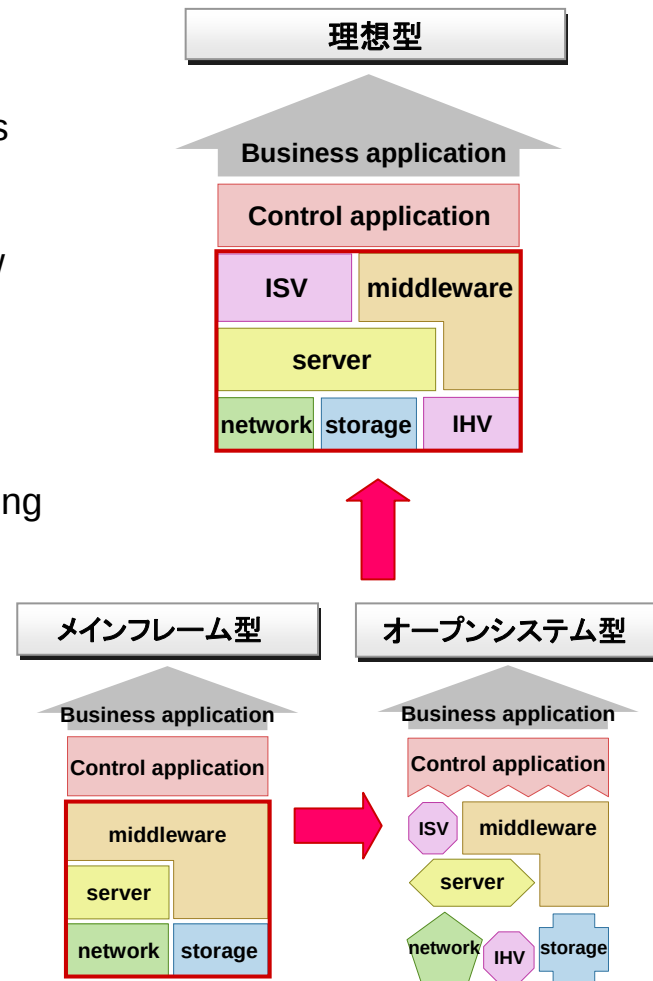
- Separates business applications from dedicated hardware
- Enables systems to respond flexibly to changes in processing loads

■ 自律 Autonomous

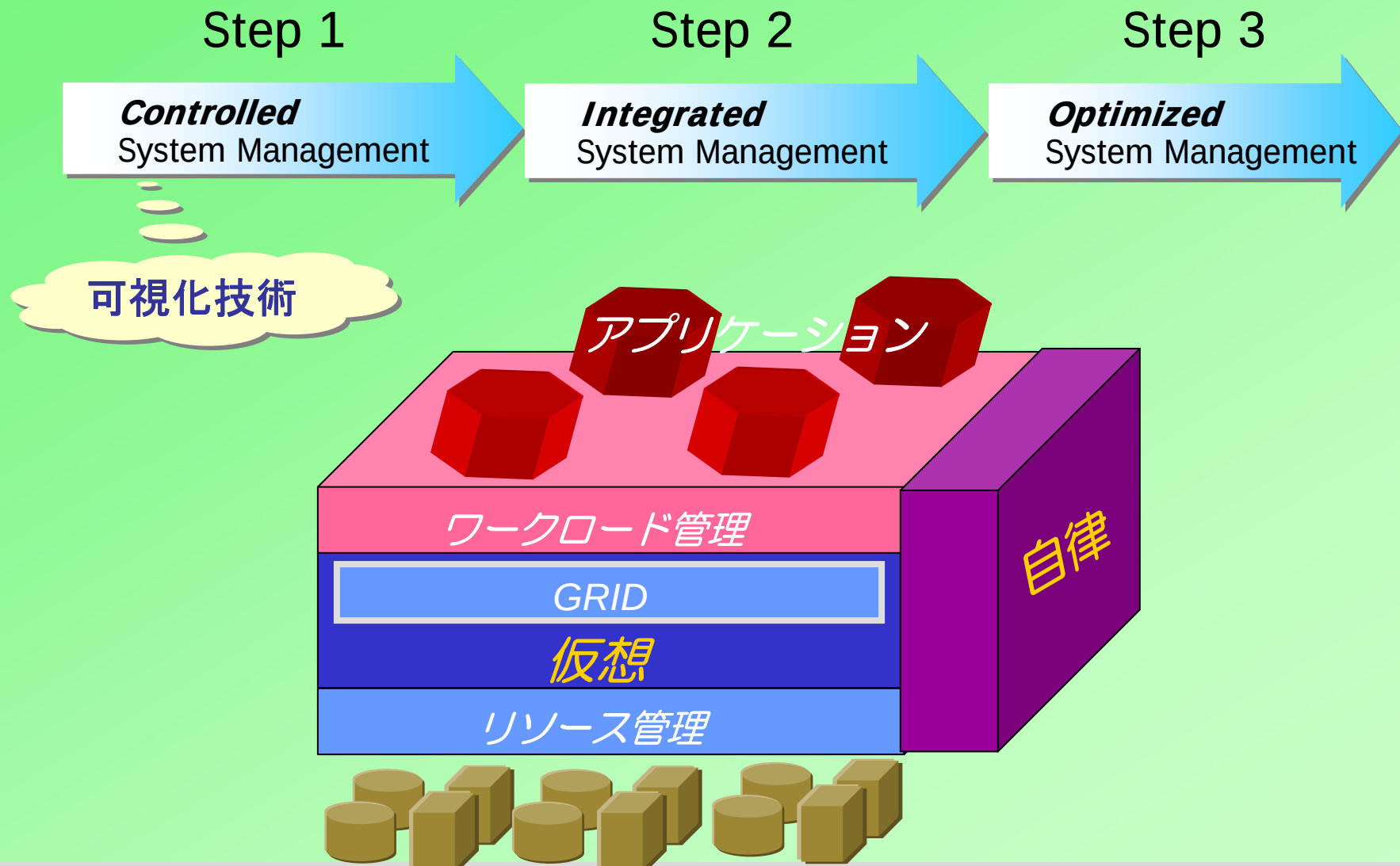
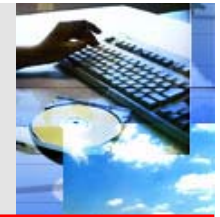
- Supports construction of self-monitoring and self-rectifying systems
- Automatically adjusts infrastructure to changes in operations and operating environment

■ 統合 Integration

- Uses validated templates to integrate processes and services
- Uses reusable, replaceable platform building blocks

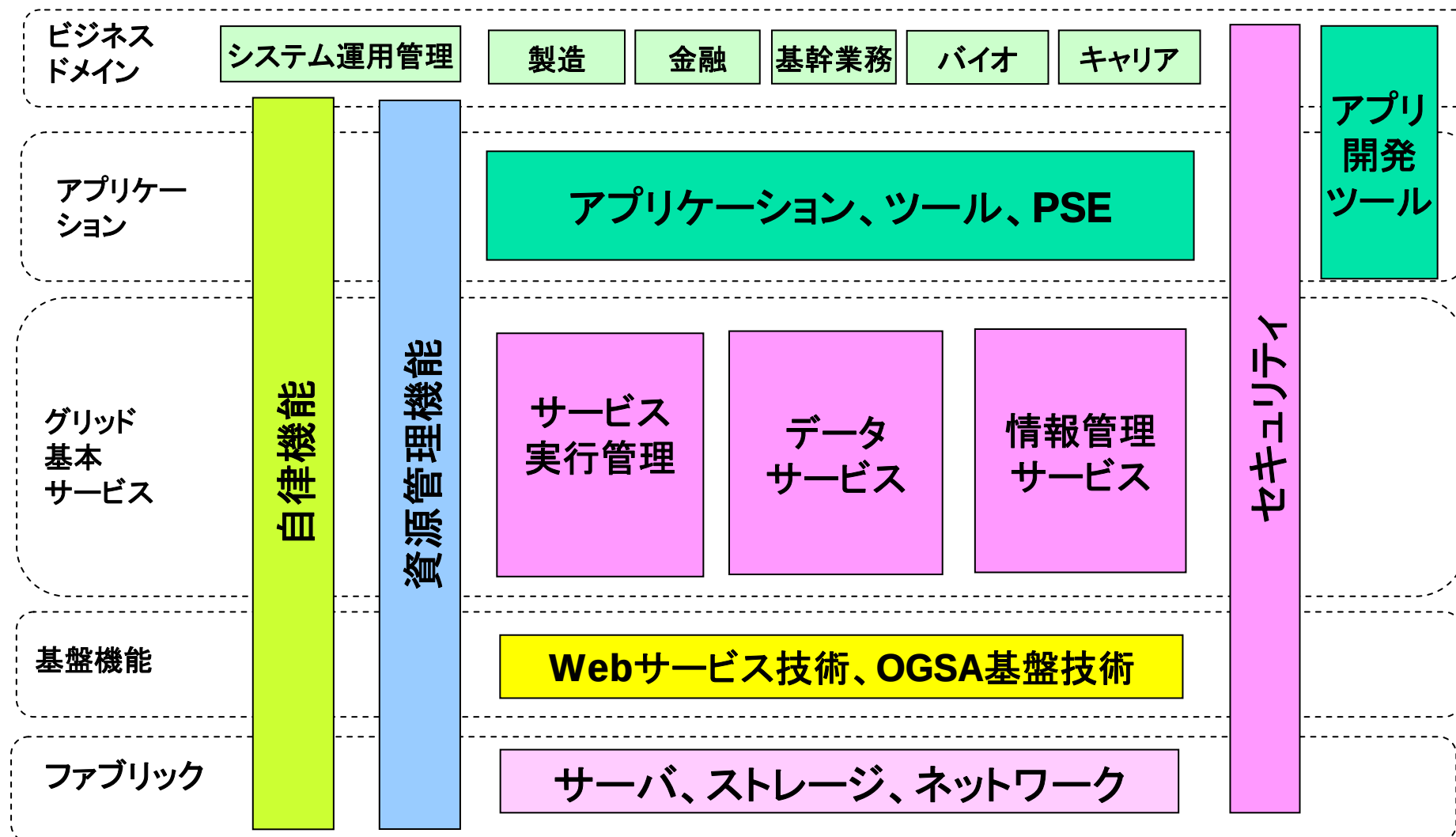


分散システムの運用

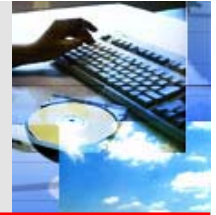


分散コンピューティング基盤アーキテクチャ

(参考: OGSA)



ニーズ



サイロから統合分散システム基盤へ

■ A社

次々にサービスを提供していかなければならないが、一方現状はサービス(アプリケーション)毎にサーバーの運用やアプリの保守をしなければならず、統合基盤のなかでサービスを運用管理したい。

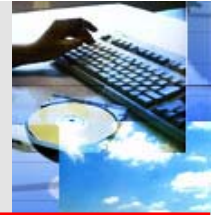
■ B社

既存のサービス毎のサービスプラットフォーム(silo)で、資源および運用管理コストが増大。新たなサービスおよび運用管理基盤を構築し、新サービスだけでなく既存サービスのマイグレーションを実施したい。

■ C社

現状のグリッド製品は組み合わせて使えない。グリッドミドル自身のサイロを壊してコンポーネントベースにしたい。

ニーズ(続)



- HPC Grid (従来からのいわゆる科学計算のGrid)
- CAD Grid
 - CADシミュレーション向けのGridを構築し、計算資源の有効活用、設計期間短縮および工数削減を図りたい。
- 事務処理バッチGrid
 - COBOLベースの事務処理バッチを汎用機からグリッド環境に置き換えたい。大規模データ処理のGrid化。
- データセンタGrid
 - リソースを仮想化し、業務(アプリケーション)から分離したい。既存システムのサイロを無くしたい。

例) 自律仮想の関連製品



【利用者の視点】

コンピューティンググリッド データグリッド
コラボレーショングリッド

 **Systemwalker CyberGRIP**

ネットワーク上に分散しているサーバを共有し、必要な時に必要なだけ提供

- 大量の計算処理を分散実行し、より速く、高い精度の処理を実現
- バッチ業務のサーバ故障トラブルを対処
- 新旧のリソース群を有効活用

【IT資源管理の視点】

データセンタグリッド
サービスグリッド

 **Systemwalker
Resource Coordinator**

ITシステムの全体最適化を実現するためのリソース制御ミドルウェア

- システムを構成するサーバ、ストレージ、ネットワーク等のリソースを一元管理
- 構成や状態に加えて、トラブル発生時の影響や原因を可視化
- 導入や構成変更、トラブル時の対処、定期保守などを自動化
- 作業における運用管理者の判断や介入に伴う負担を軽減

3. システム・セキュリティ



外部からの脅威対策

内部の漏洩対策

マネジメント、
コンプライアンス対応へ

2000年

不正アクセス
官公庁、jpドメイン
無差別攻撃

不正アクセス
禁止法

電子署名法

ISMS認証取得 事業者数 : 1,548
～2006年4月21日現在～
プライバシーマーク認定事業者数 : 3,736
～2006年4月24日現在～

2004年

ワーム被害
CodeRed, Nimda
SQL Slammer

相次ぐ
情報漏洩事件

脆弱性情報取り扱い
(WebAPLも対象)

情報セキュリティ
監査制度

2005年

Phishing
Spyware

ISO17799
改定
→2005年5月

個人情報保護法
e-文書法

2006年

政府機関の
セキュリティ統一基準

重要インフラにおける情報セキュ
リティ確保に係る『安全基準等』
策定にあたっての指針

ISO27002
(ISMS実践規範)
→JIS Q27002(予定)

ISO27001(05/11)
(ISMS要求事項)
→JIS Q27001

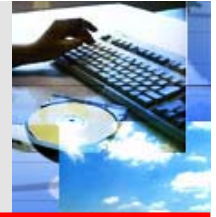
保証型セキュリティ
監査本格化

2007年

BCP ISO化
(2007-2008)

金融商品取引法
(日本版SOX)
(2008)

新たなセキュリティ・ニーズ



- 様々な事件、規制緩和／新たな法制度、等々により、セキュリティから法令遵守やCSR、リスク管理という観点でのビジネスニーズが一気に拡大



企業内部統制の視点でのITニーズ急増
セキュリティ技術もコンプライアンス/内部統制のための総合対策へ

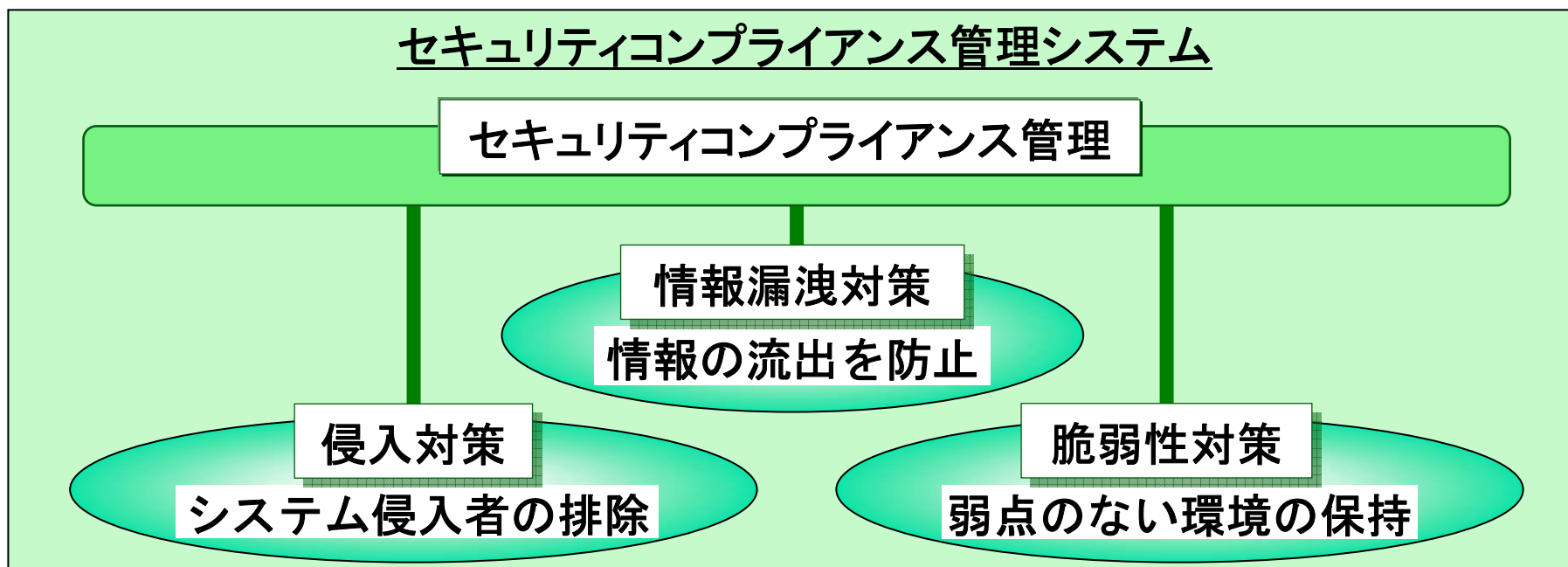
コンプライアンスのための管理体系



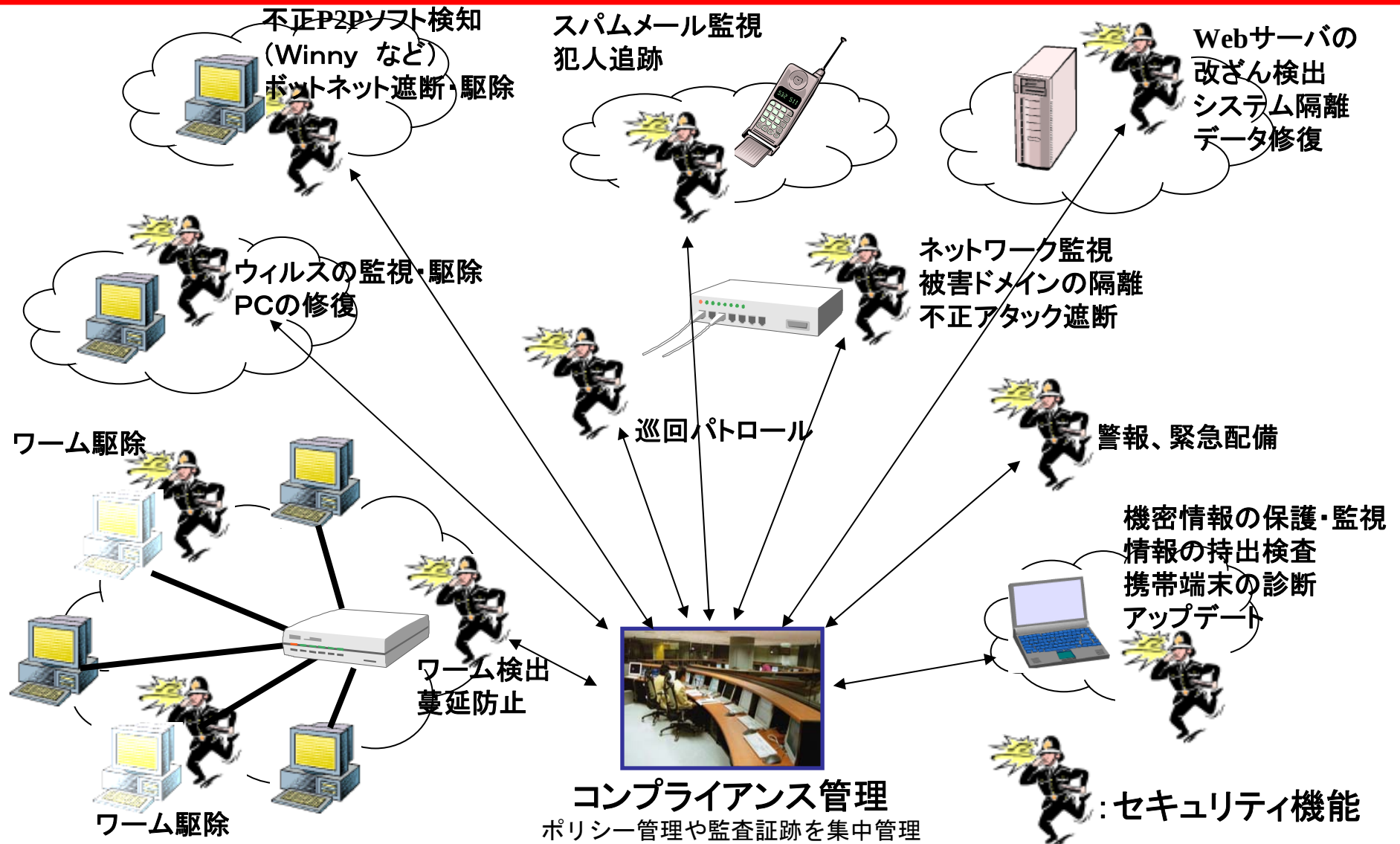
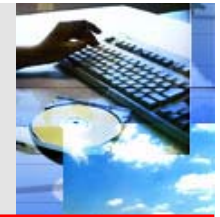
- ベンダーは、システムセキュリティを実現するソリューションを提供

例) 富士通のセキュリティコンプライアンス管理体系

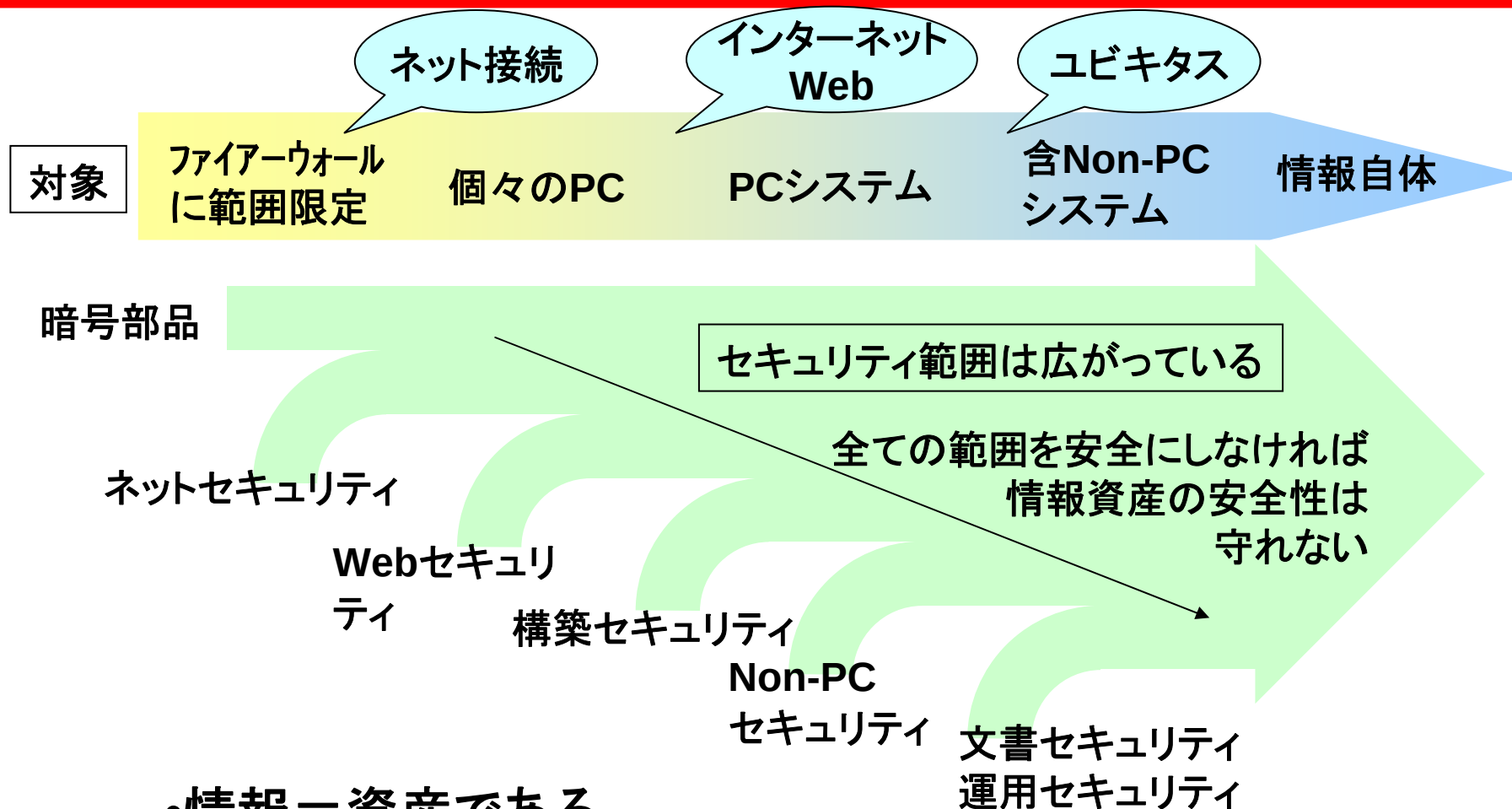
- 情報漏洩を防止する機能を3つに分類:
 - 侵入対策 システム侵入者の排除
 - 脆弱性対策 弱点のない環境の保持
 - 情報漏洩対策 情報の流出を防止
- 上記の対策機能を統合的に管理:
 - セキュリティコンプライアンス管理



コンプライアンス管理イメージ

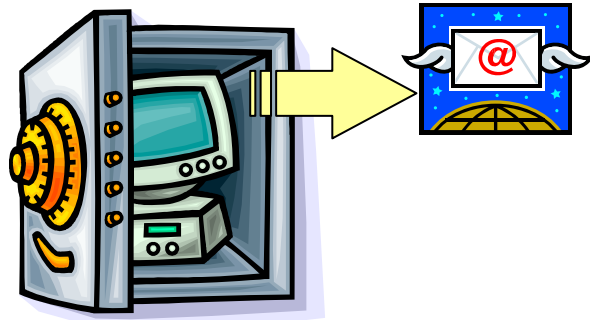
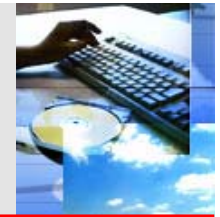


情報漏洩のための対象機器の拡大



- 情報＝資産である
- 情報資産を守るためにシステムをセキュアにしてきた
- 今後は情報に注目することが必要

次世代の情報保護

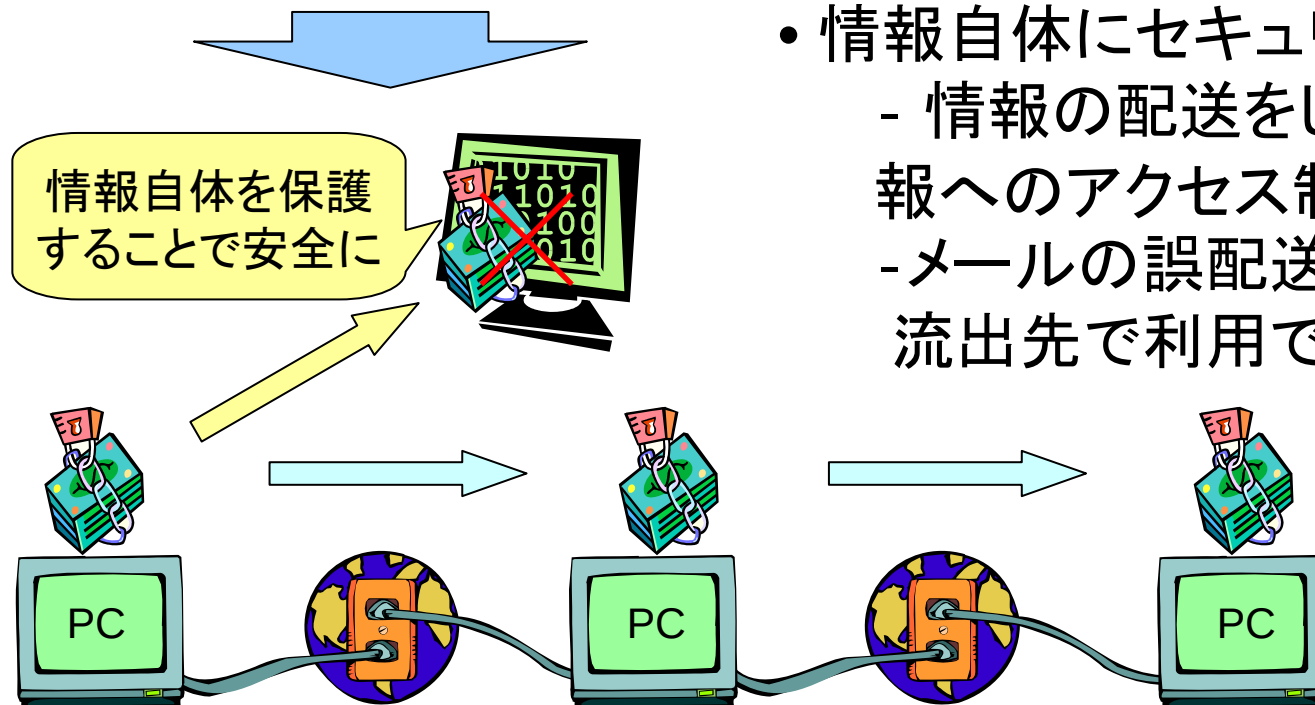


■従来の情報管理

- PC/システムを金庫に入れて守る
 - PC/システム内は安全
 - 外に出た情報は防御できない

■新たな情報保護の枠組み

- 情報自体にセキュリティを付加し保護
 - 情報の配送をしても配送先での情報へのアクセス制限可能に
 - メールでの誤配送/故意の流出でも流出先で利用できない

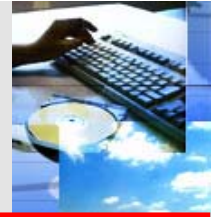


セキュリティの方向性



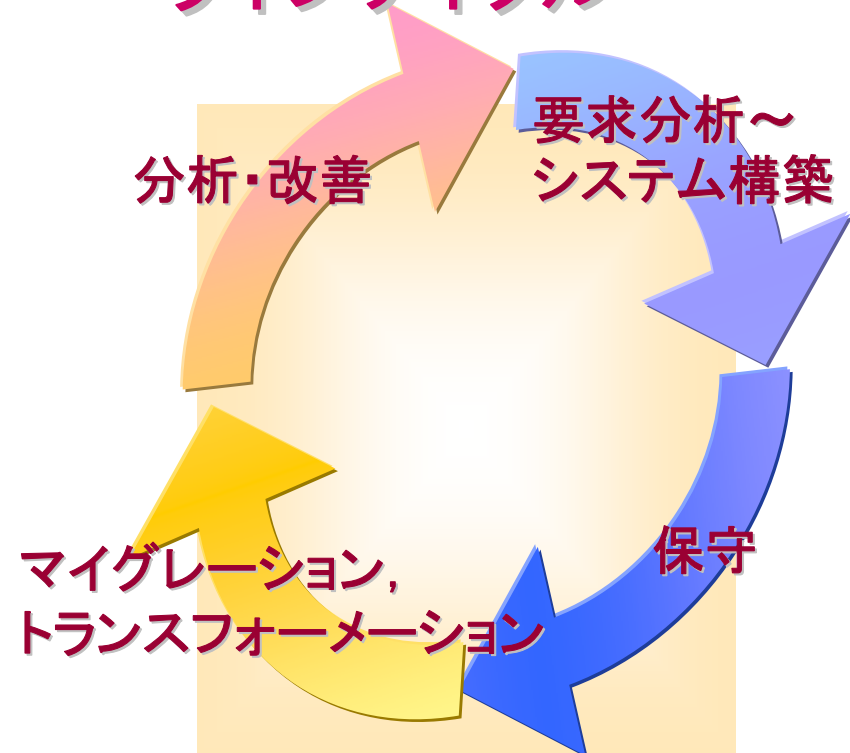
- 企業ブランドの観点から
 - 企業内部統制やコンプライアンスが重要に
- システム構築の観点から
 - 総合的なコンプライアンス管理を可能とする機能
 - システム構築段階からセキュリティを作りこむ
 - ◆ 開発段階でのセキュリティ機構の組み込み
 - PDCAサイクルによるシステムセキュリティの向上
 - ◆ 現状のシステムの状況を把握できる
 - ◆ 情報セキュリティ対策による変化が把握可能

4. ビジネスの為のソフトウェア工学

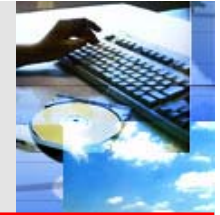


- BPMのための環境
 - * Business Process Management
- 要求工学
- システムやIDC運用まで含めた品質向上
- 既存システムを含めたシステムの保守拡張

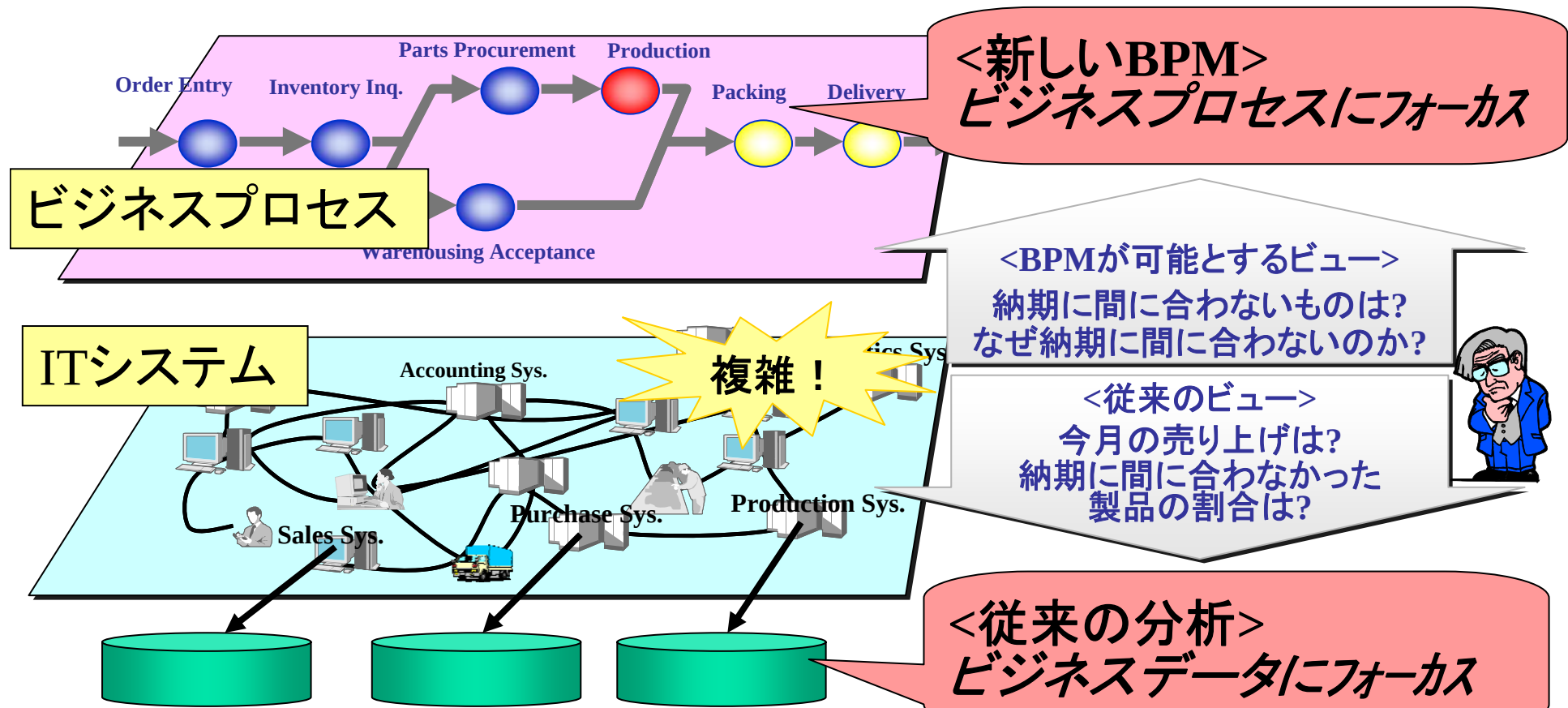
ソフトウェアシステムのライフサイクル



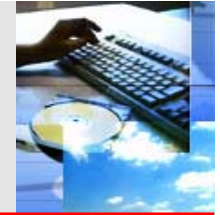
BPMの重要性



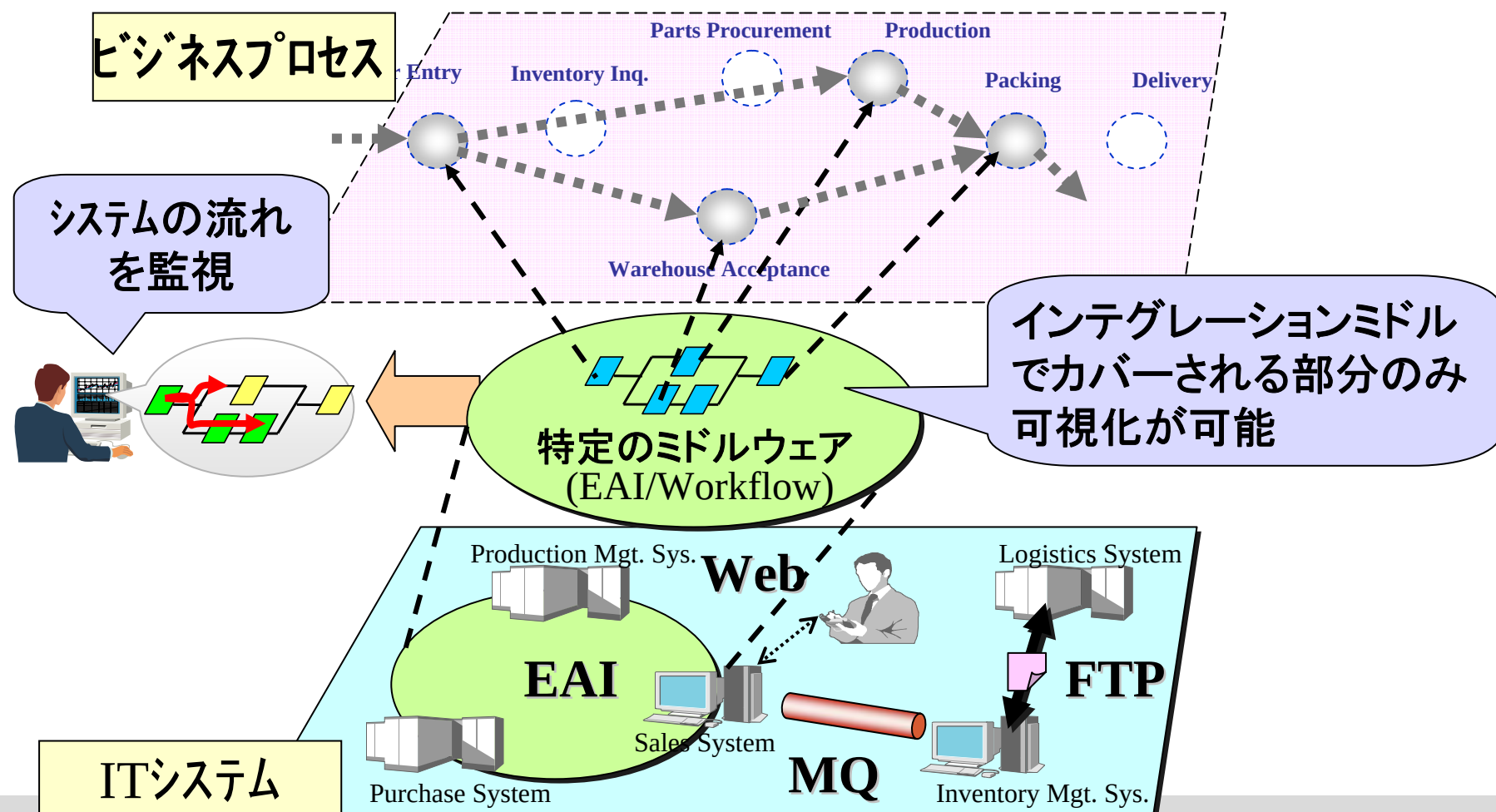
- 複雑なITシステムはビジネスの問題点を隠蔽してしまう
- ITシステムから見たビジネスプロセスは問題の解決に役立つ



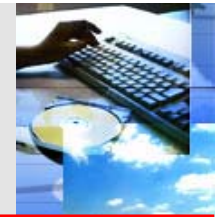
従来のBPM/BAM技術



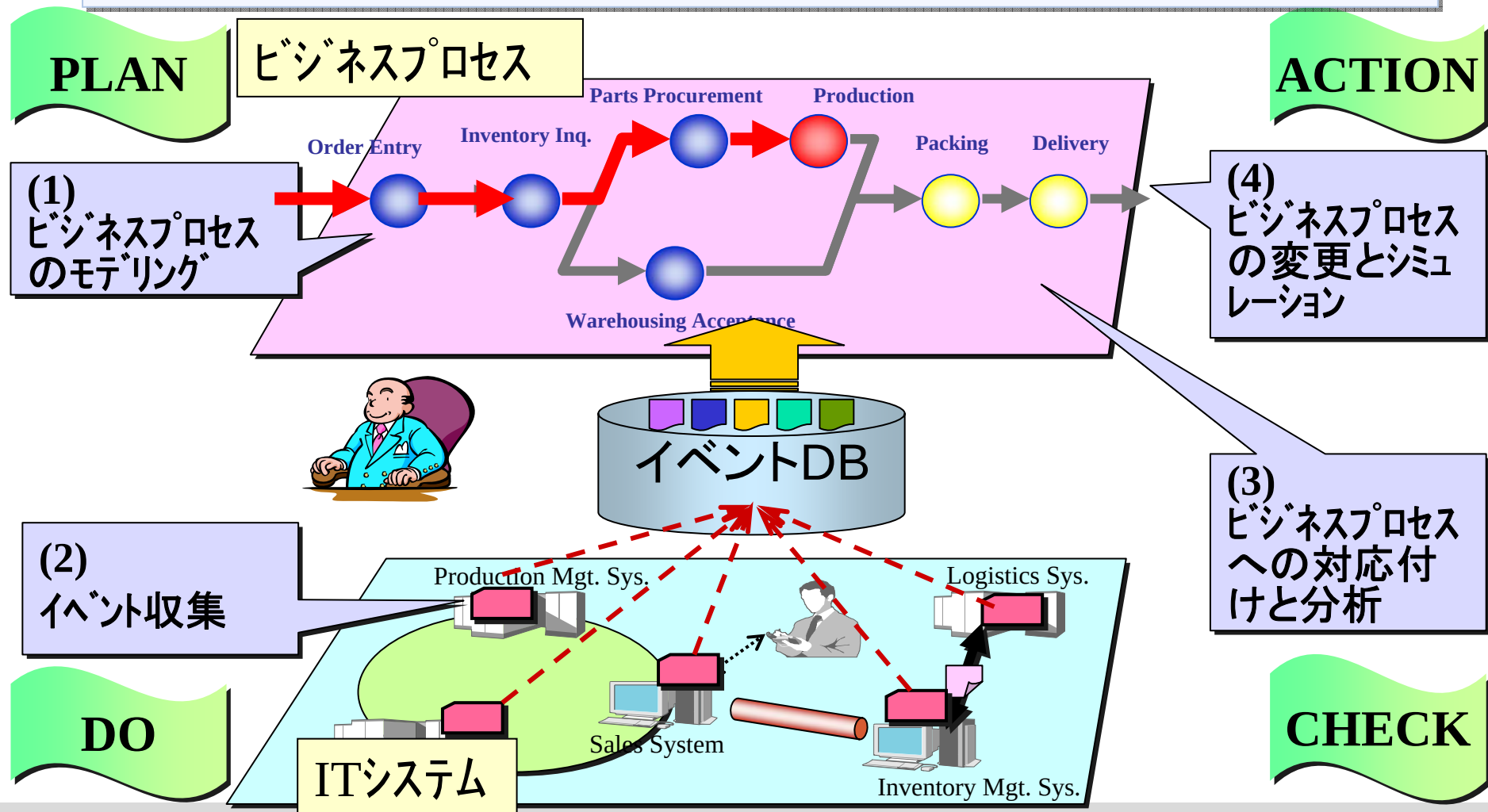
特定のインテグレーションミドルウェア上のシステムのみ可視化するため、他システムを含む全体プロセスの可視化は不可能。



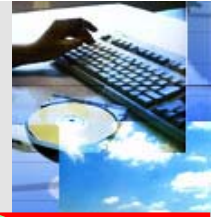
新しいBPMのビジョン



ITシステム全体の可視化を可能とすることで、ビジネスプロセス・レベルの改善を容易とする



新しいBPMのための技術



PLAN

ACTION

技術1:
ビジネスプロセスの
モデリング

技術4:
プロセス改善のための
シミュレーション

ビジネスプロセス

技術2:
システムの変更無し
でのイベント収集

技術3:
イベント分析による
プロセス可視化

イベントDB

DO

CHECK

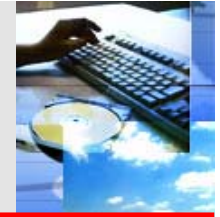
ITシステム

Production Mgt. Sys.

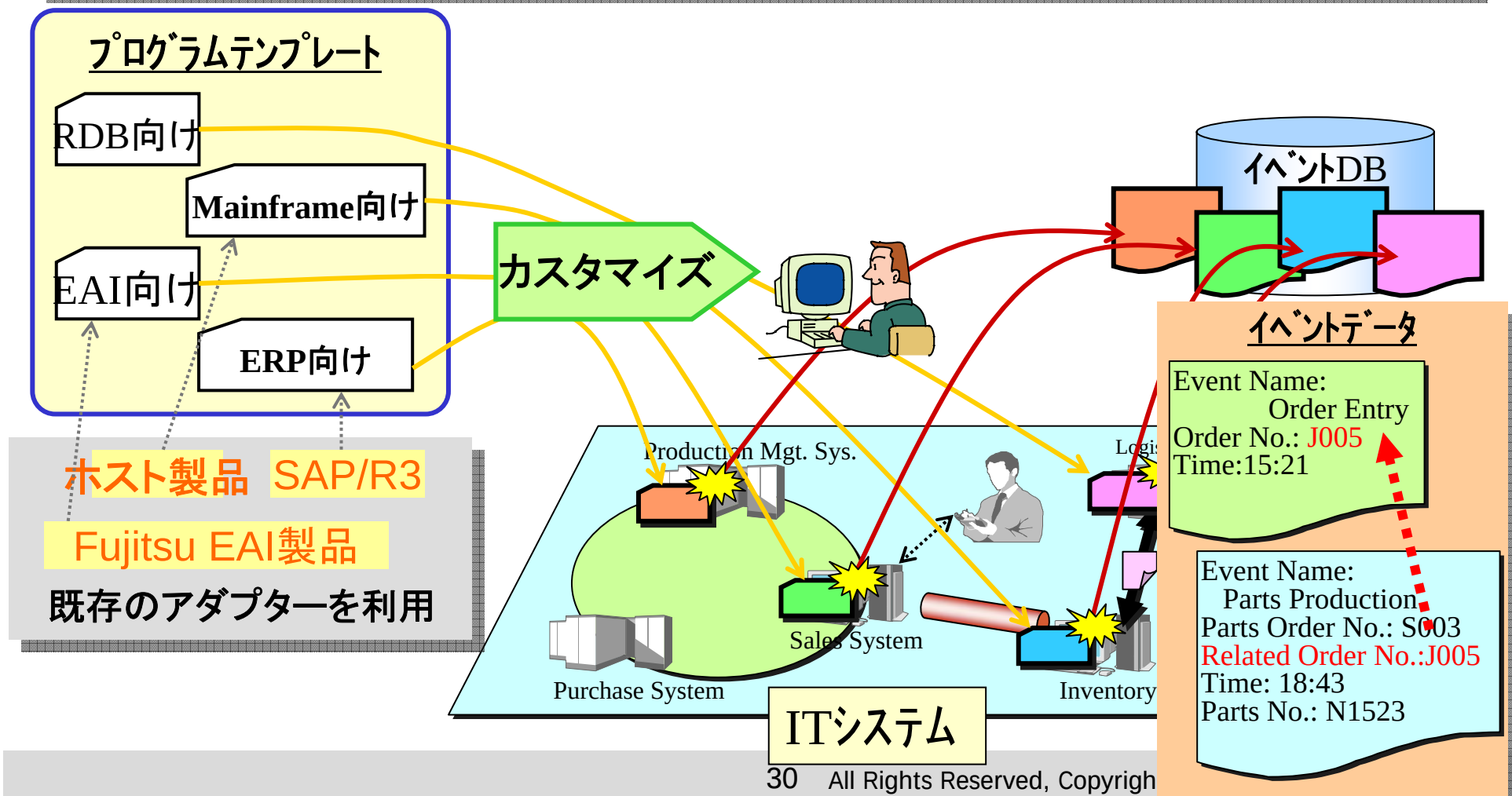
Sales System

Inventory Mgt. Sys.

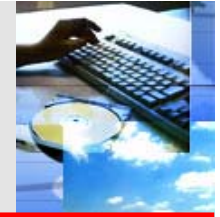
例) 技術2: イベント収集



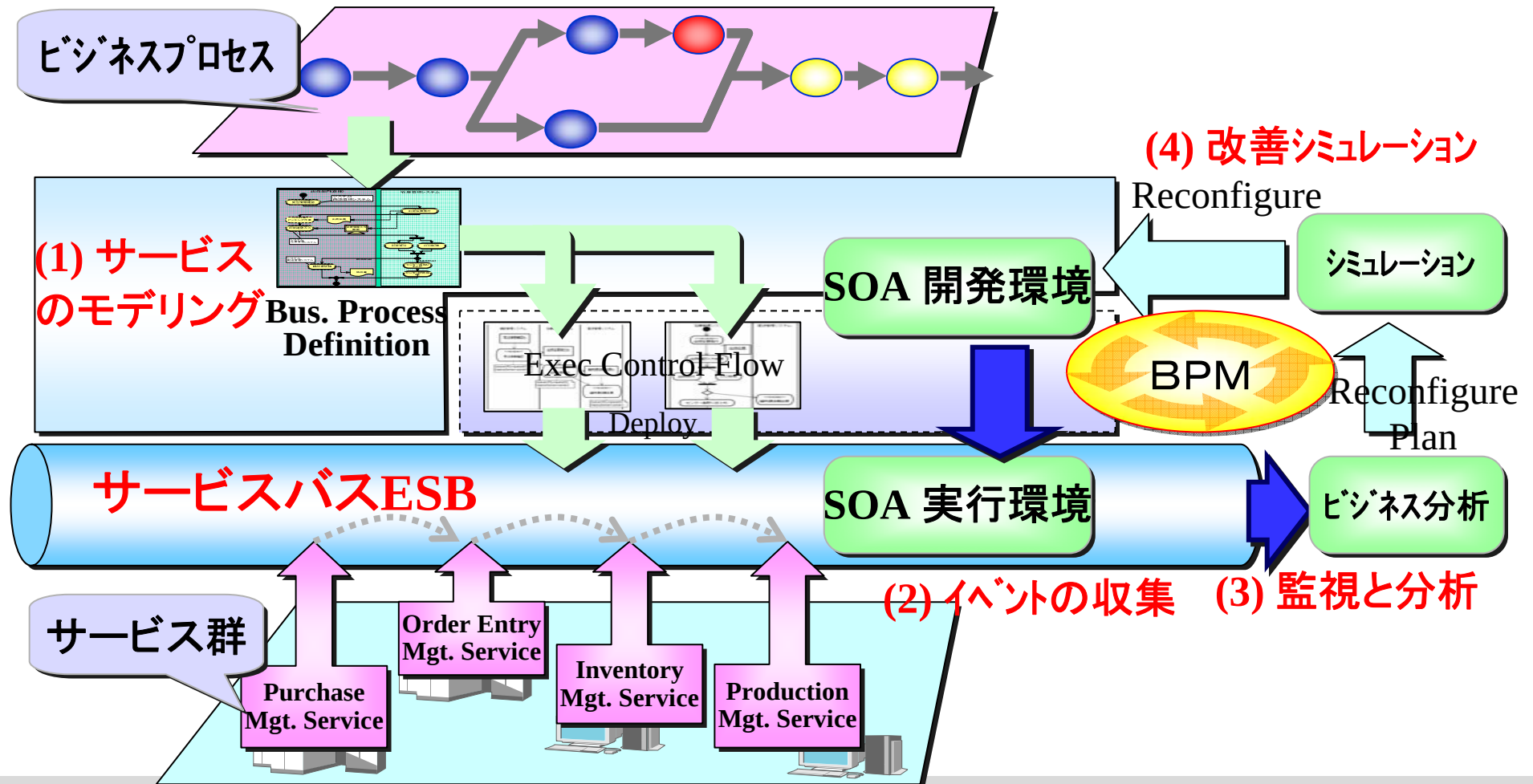
- システムからのイベントの抽出と関係把握
- 複雑なイベントデータを抽出するプログラムのテンプレート化・自動化



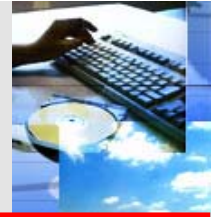
SOAによるBPMの強化



- SOAのESBを, BPMのPDCAサイクルを実現すべき強化



5. おわりに



- システム化とビジネスとのリンクは益々重要となり、そのための(利用)技術が発展
- ネットワーク上のリソースは自由に扱えるようになる
- ソフトウェアのコンポーネント化と統合技術が進む

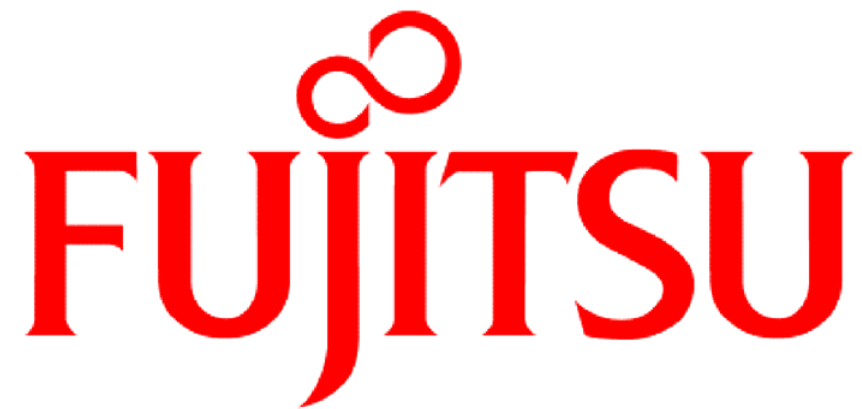
ビジネス

新しいビジネスモデルとIT技術への要求！

ソフトウェアシステム技術

↑ Supports

複雑で優れた技術が求められる！



THE POSSIBILITIES ARE INFINITE