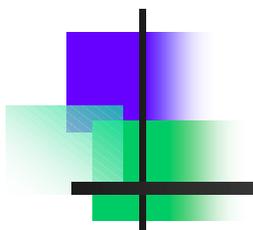




XML Consortium



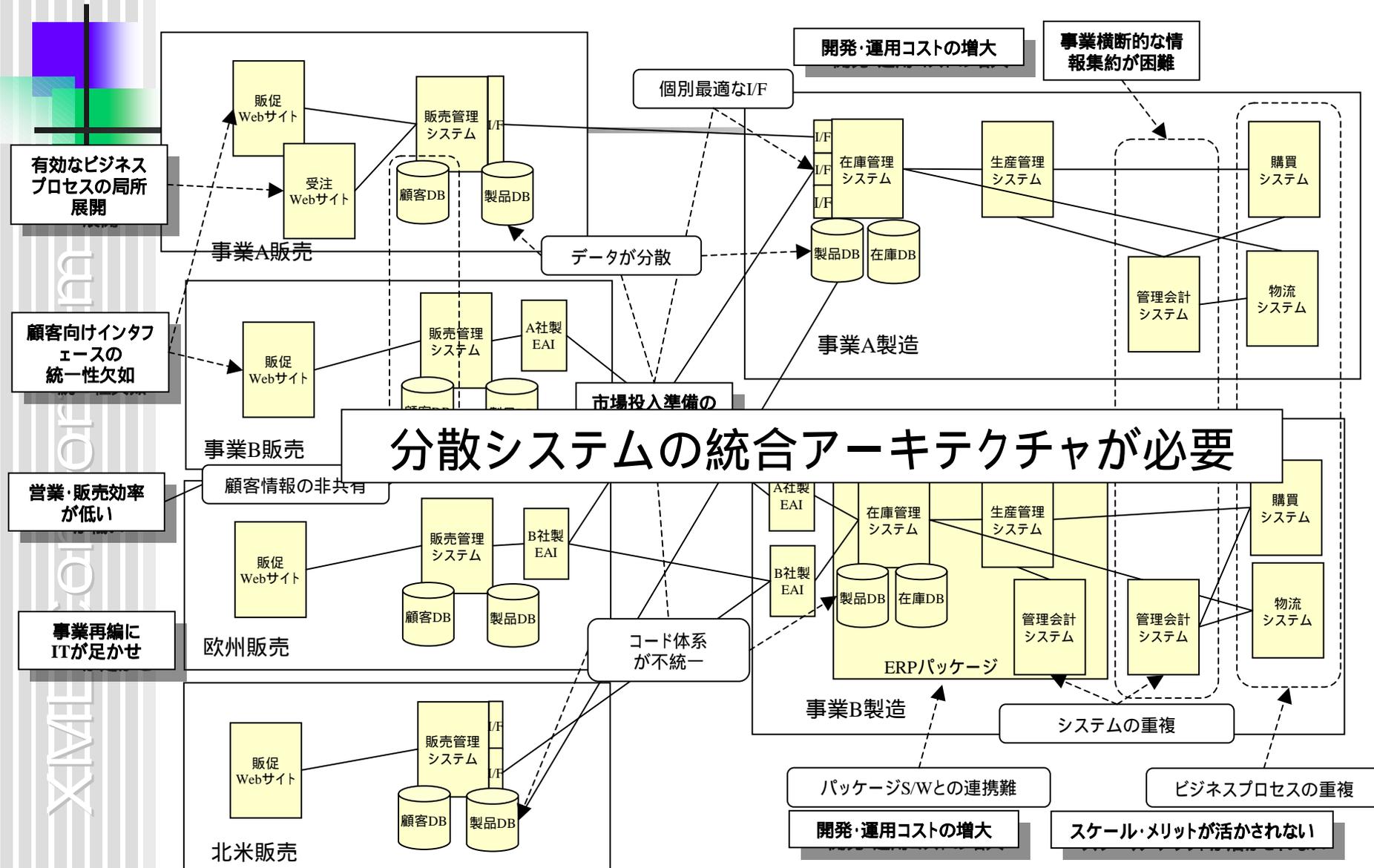
SOAとは？

2005-10-19

ビジネスイノベーション研究部会 牧野友紀

- 1 **なぜSOAが必要か？**
- 2 **アーキテクチャ概念**
- 3 **技術アーキテクチャ**
- 4 **モデリングのポイント**

分散システム環境の問題



SOAが求められる背景

運用面での経済性、業務変革に対する柔軟性、企業・組織間業務連携の効率性、即時性、の要求に対して、従来の分散システム環境の課題が多く見受けられるようになってきている。

課題解決

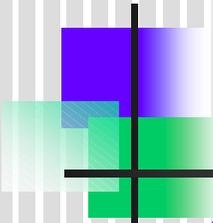
要求事項

- ビジネス変革とIT革新への**柔軟な適応性**
- 資源(人、システム、コスト)の**全体最適化**
- システム機能の**部門最適化**
- 部門間のビジネスプロセス/情報の**オンデマンドな統合能力**
- 全体システムの**一貫性の向上と維持**

実現手段

実現方法

- ビジネス戦略にITを関連づけるメソドロジー
- 分散システムを構成要素とする統合システムのアーキテクチャ
- 将来像とロードマップを示すランド・デザイン参照モデル



SOAとは

- 散在するシステムを、業務視点の機能(サービス)の集合と捉え、ビジネス環境の変化に対して、迅速にサービスを組み換える (Business Process Integration) ことで、柔軟に対応するシステム構築方針
 - アプリケーションをラップしてサービスにする
 - 外からはサービスのインタフェースのみを意識する
 - ハードウェア/OS/ミドルウェア
 - 開発言語 等々を隠蔽
 - サービス自体の実装には依らない
 - インフォメーションバス (ネットワーク)にサービスをプラグインして統合を実現する
- 具体的には上記を実現するための
 - アーキテクチャパターン
 - デザインパターン
 - ベストプラクティス
 - 等々の集まりである

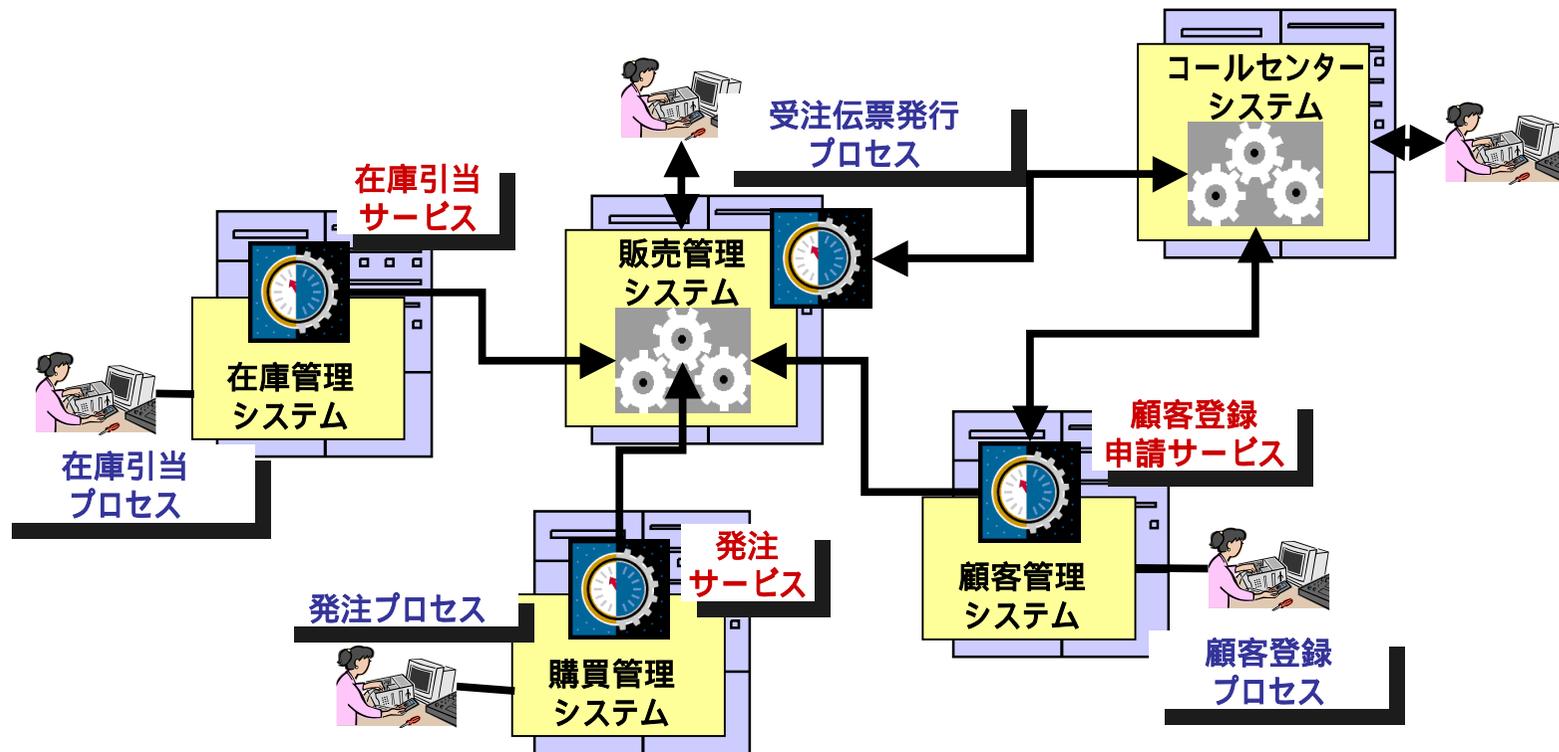
SOAのモジュール化概念

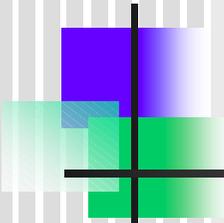
- **アーキテクチャ**
 - 分散システムをモジュール化し、構造的に統合する概念
 - ビジネスプロセス・モデルの構造を写像
 - 各システム的设计、実装、運用の自由度を維持
 - モジュールの入れ替えを容易にするインタフェースと実装の分離、標準プロトコルによるデータ交換
- **インタフェース**
 - モジュールの入力データ構造と出力データ構造、データ転送方式の明確化
 - 実装非依存
- **標準**
 - インタフェース記述言語の標準化
 - データ型指定、メッセージ形式、メッセージ交換規約の標準化
 - Webサービスの利用が主流

SOAベースのシステム構成

関連システムが提供するサービスを組み込み業務連携の自動化を実現する。

また、新規システム開発に、既存システムのサービスを利用し生産性と品質を向上させる。



A decorative graphic consisting of a vertical line and a horizontal line intersecting at a point, with a purple square above the intersection and a green square below it.

今までと何が違うのか

- SOAという概念は以前からある
 - 概念としてはシンプルで説得力がある
 - しかし、期待したほどの成果はあがっていない
- Corba, DCOM, EJBに足りなかったもの
 - 共通のインタフェース定義言語
 - 共通のインフォメーションバス
 - 柔軟なインタフェース記述
 - インタフェースとプログラミングモデルの分離

WebサービスがSOAにもたらしたもの

- Webサービスはインターオペラビリティにフォーカス
 - 共通のインタフェース定義言語
 - WSDLによるインタフェースの記述
 - 共通の情報メーションバス
 - インターネットプロトコルによるEnd-to-Endの接続
 - 柔軟で抽象度の高いインタフェース記述
 - RPC->ドキュメントベースの記述へ
 - Simple Endpoints, Rich Dataのスキームへ
 - インタフェースとプログラミングモデルの分離
 - 疎結合(ミドルウェアへの依存性の解消)

なぜ重要なのか？ソフトウェア開発の生産性だけではない。

■ 企業のIT関連最大関心事はビジネスプロセスインテグレーション

Webサービスは、SOAPやWSDLなどのテクノロジーの総称でもありますが、また、ビジネスのイノベーションのあり方に対するコンセプトでもあります。また、このコンセプトは十分市民権を得ているわけではないようですが、Webサービスの本質を理解する上で重要です。

今まで、情報技術は、ビジネスプロセスを効率化することで企業の競争力の向上に寄与してきました。最初は会計処理などの定型業務から始まり、オフィスなどの非定型業務もITなしには行えないようになってきました。現在のIT利用の主流は、SAPなどによる、企業全体のビジネスフローの効率化に移ってきています。個々の業務だけではなく、それらを統合化した形での企業の効率化を行うわけです。しかし、これらはいずれも、「競争相手と基本的に同じ業務フローを、効率化によって差別化する」という企業戦略に基づいています。

Webサービスの根底に流れるのは、「競争相手がまだやったことのない、新たなビジネスフローを作り出すことによって差別化する」というビジネス戦略です。今まで誰もやったことのないビジネスフローを作り出すのは、どうしたらよいでしょうか？それには、型にはまらない考え方が必要です。かといって、個々のビジネスプロセスをゼロから設計するのでは、この動きの速い世界で他人より先に出ることは難しいでしょう。Webサービスのコンセプトは、「社内の個々のビジネスプロセスを部品化しておくことによって、それらの異なる組み合わせをいつでも自由に作ることができる」ということです(ここでの「ビジネスプロセス」は、システム的なものだけでなくことに注意してください。

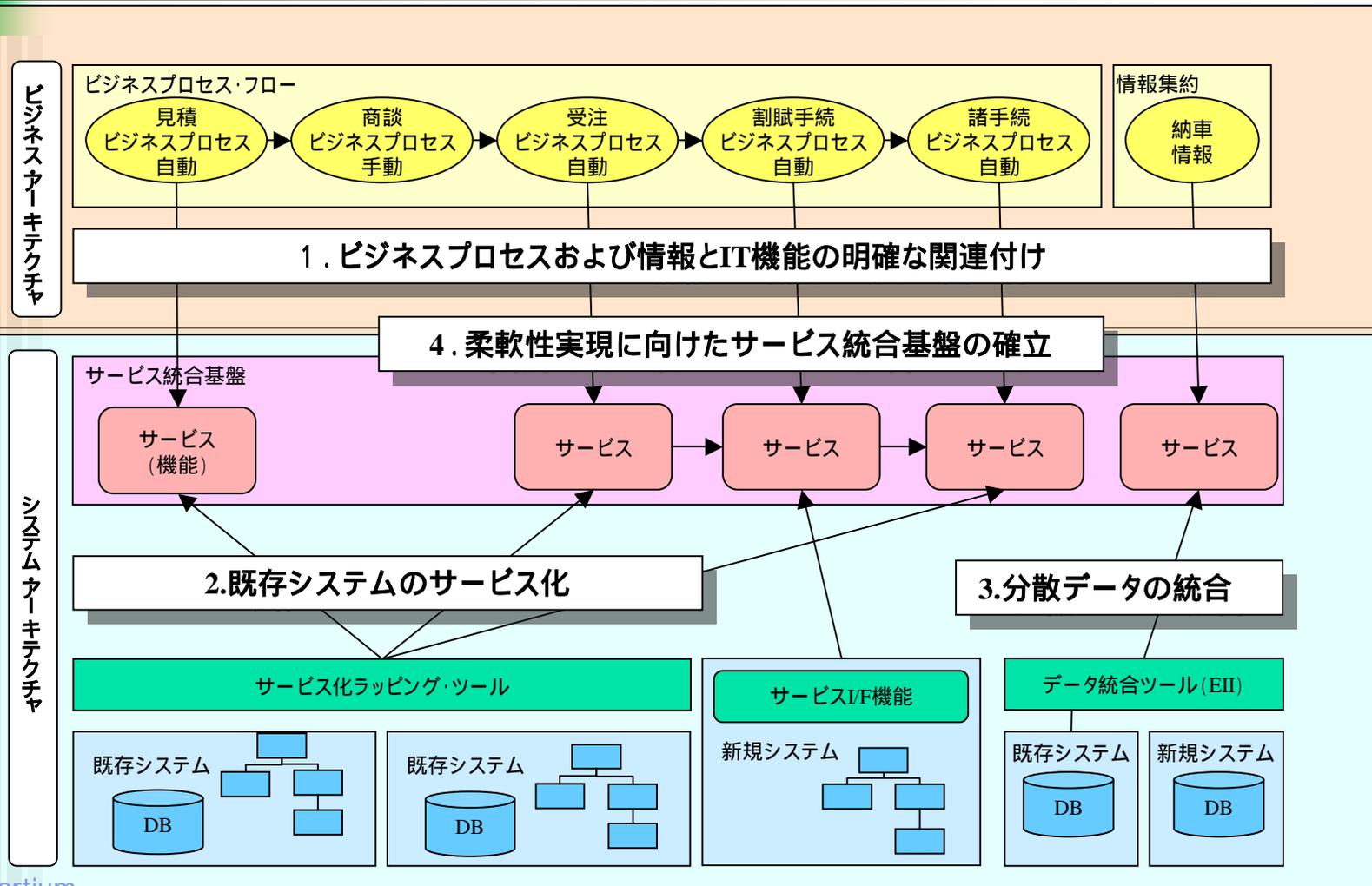
人手によるプロセスは企業にとっては、システム的なプロセス以上に重要な資産です)。このようなビジネスプロセスの部品化と再利用は、社内の業務処理の標準化などを通して今までも試みられてきています。Webサービスは、この部品化と再利用のプロセスを効率化するための、ITが提供するソリューションなのです。 Webサービスの実践は始まっていますが、残念ながら、いまだにこのような大きな企業戦略に基づいて社内業務のWebサービス化を行っている企業は少ないと思います。しかし、これこそが、Webサービスが本当に目指しているものだとは私は信じています。

(XMLテクノロジー通信7号http://www.xmlconsortium.org/wg/tech/tech_mlmaga_index.html)

日本IBM 丸山 宏氏「企業戦略としてのWebサービス」より)

ビジネスプロセスとSOAの関係

XML Consortium



1

なぜSOAが必要か？

2

アーキテクチャ概念

3

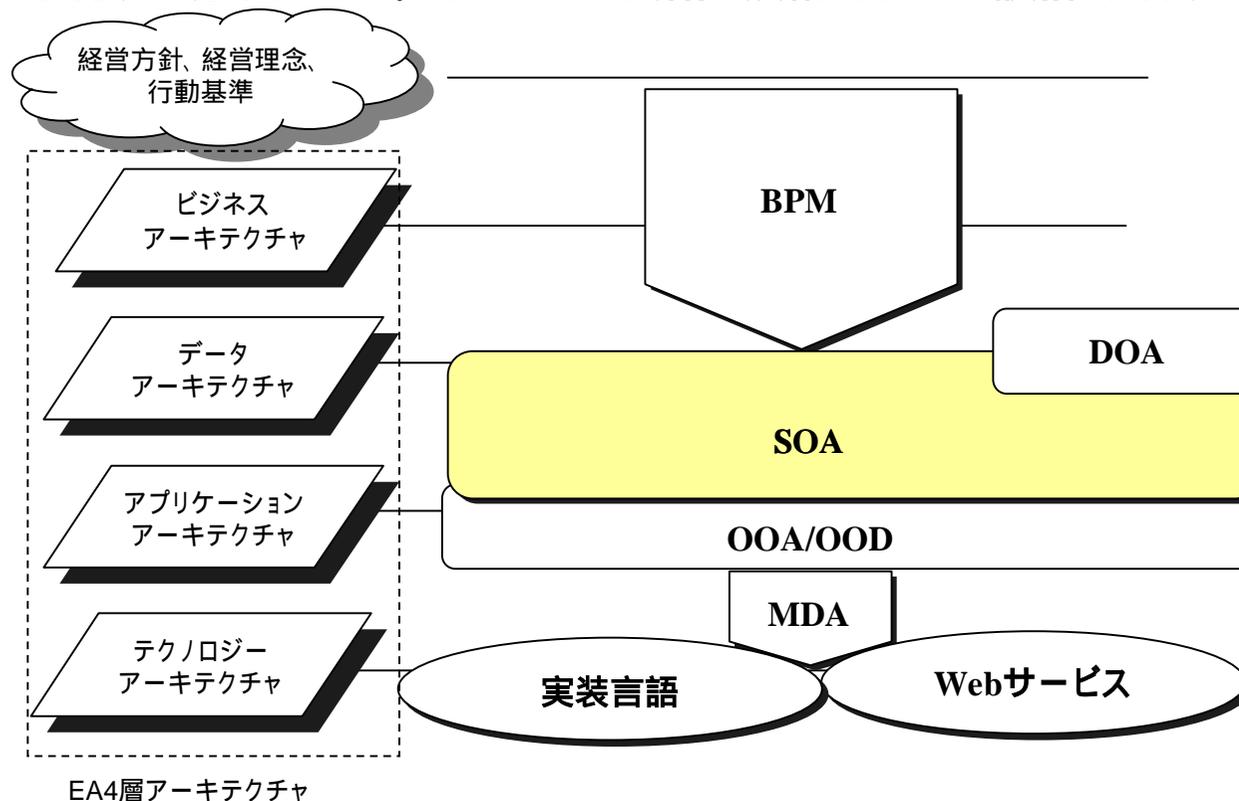
技術アーキテクチャ

4

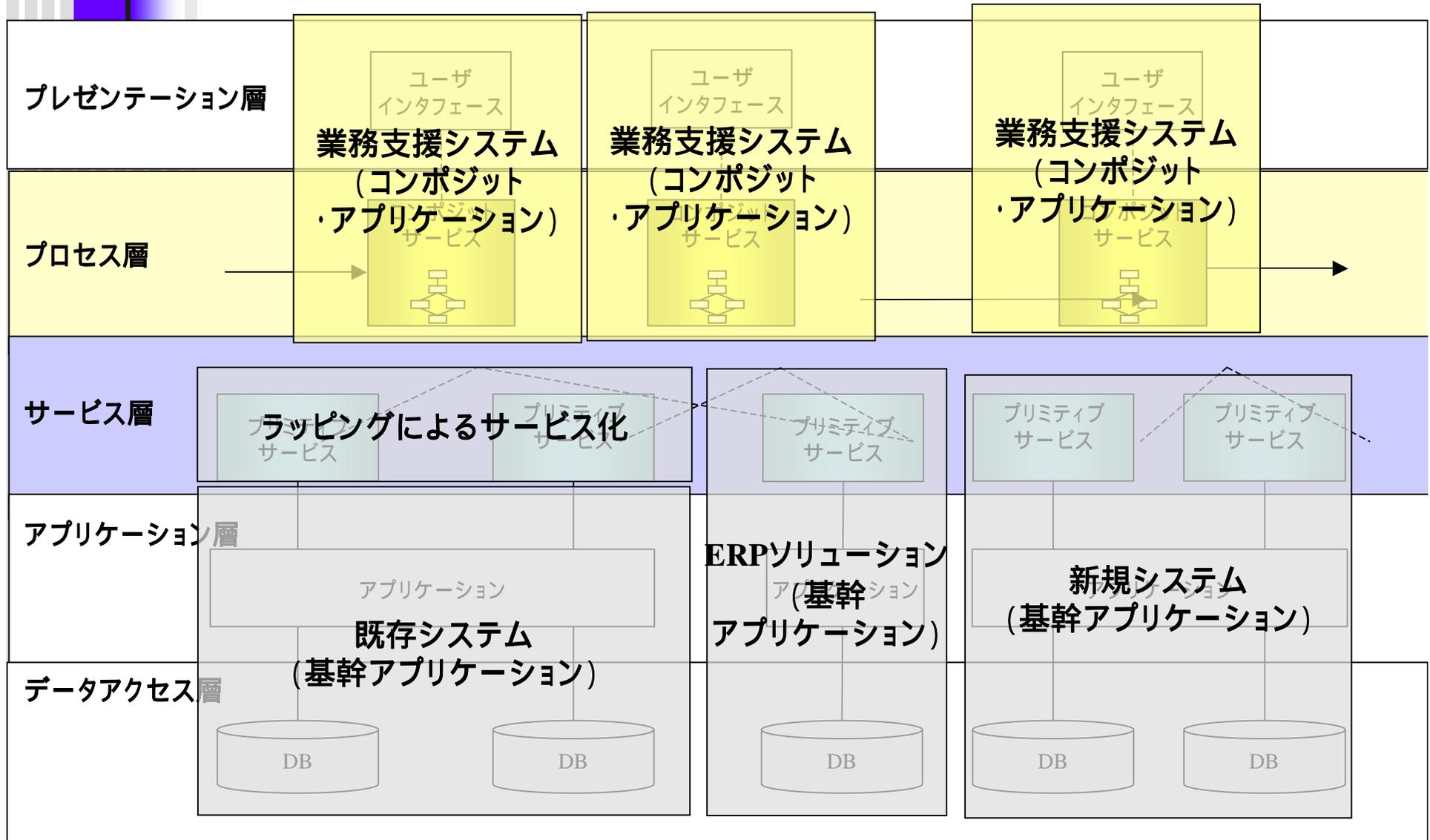
モデリングのポイント

SOAの位置づけ

SOAはデータ・アーキテクチャとアプリケーション・アーキテクチャに関連する設計原則。
 サービスの実装の規定はない。オブジェクト指向技術等により設計・実装する。



SOAの5層構造



サービス層とプロセス層

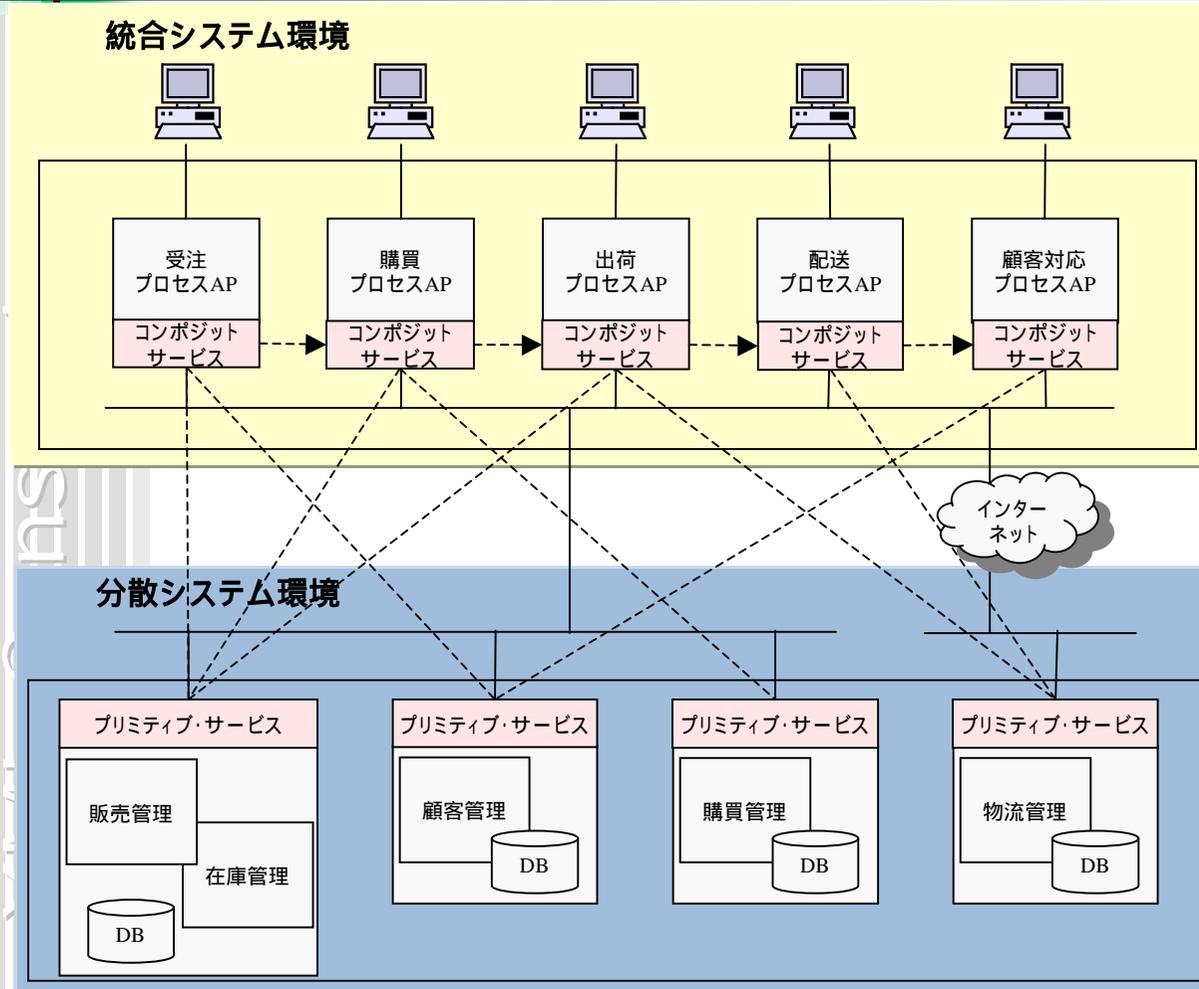
■ サービス層

- 既存システム群の機能を抽象化し粒度を平準化することを目的とした層
- アプリケーション層のコンポーネント、プログラムを業務視点で集約し実現するサービス群
- 再利用性を最大の目的としたコンポーネント型サービス(プリミティブ・サービス)
- ビジネス・エンティティ単位にモジュール化することが一般的

■ プロセス層

- ビジネスプロセス・フロー変更に対応することを目的とした層
- ビジネスプロセス・モジュール単位に関連付けプロセスを支援する機能を複合したサービス群
- ビジネスプロセスの共有に合わせ、提供されるプロセス型サービス(コンポジット・サービス)
- 複数のコンポーネント型サービスを複合する
- 内部にワークフロー制御を実装する : BPEL適用領域

SOAベースのアプリケーション例



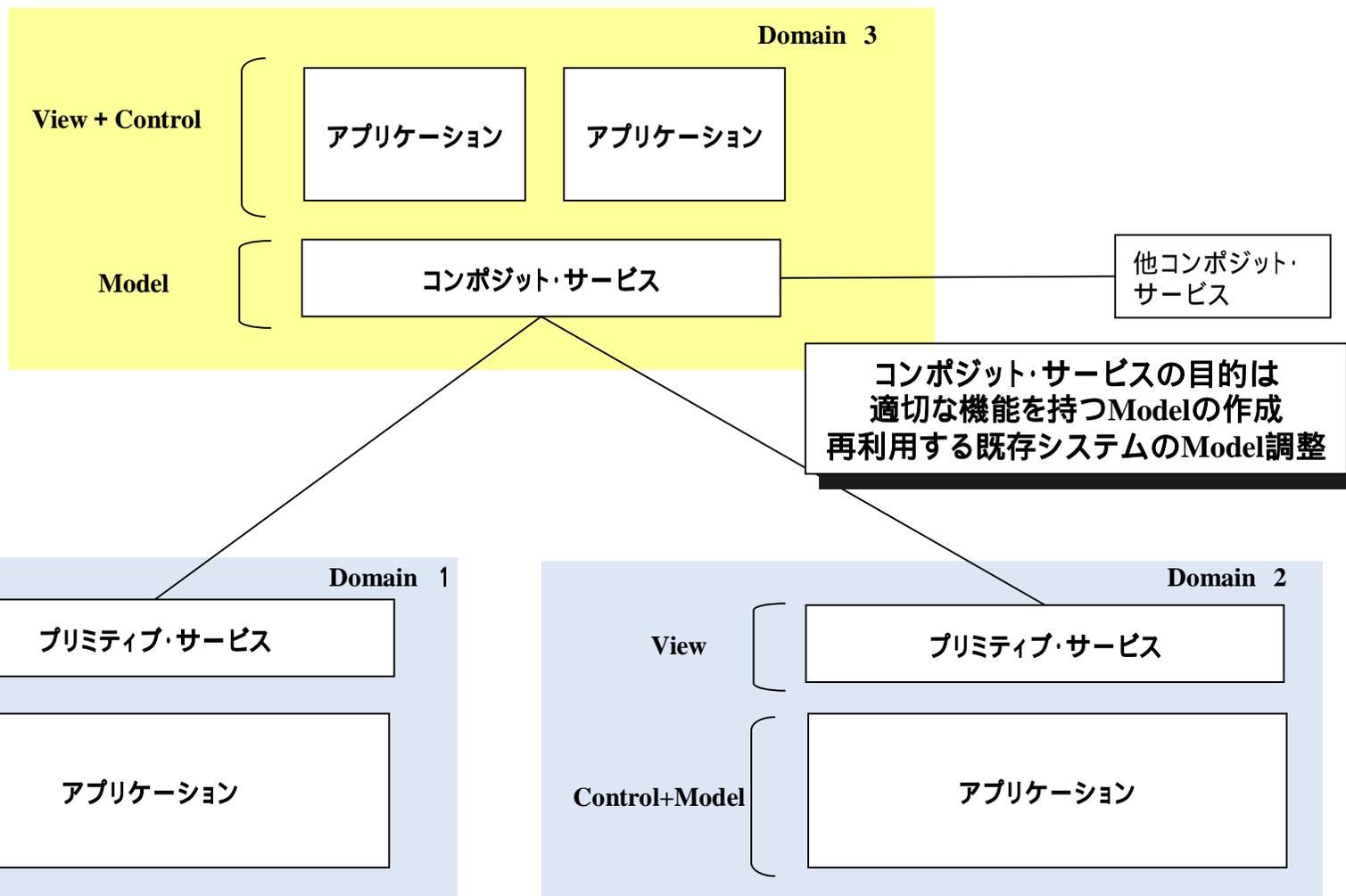
コンポジット・アプリケーション

業務組織(役割)単位ごとのシステムで、業務を支援する機能を集約し包括的に提供する。ジョブ・レベルのフロー制御、一時的なデータ管理を行う。

基幹アプリケーション

組織横断的なシステムで、基幹となるトランザクション処理、永続データ管理、バッチ処理を行う。

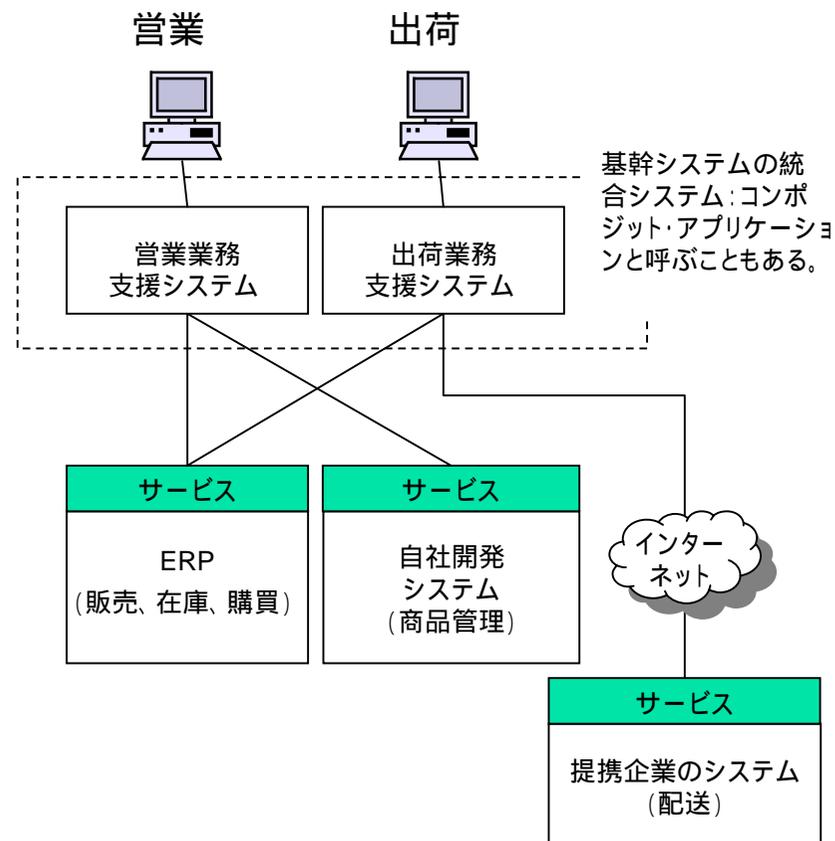
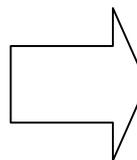
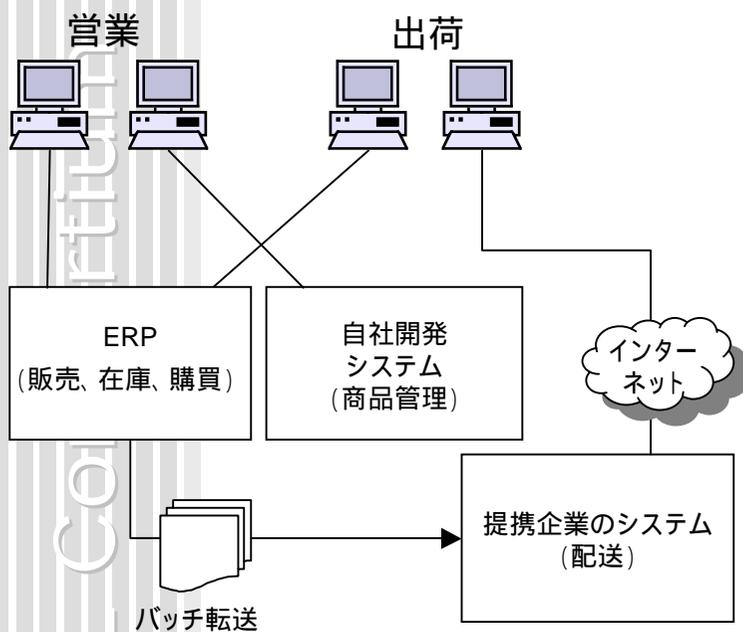
コンポジットサービスと プリミティブサービスの関係



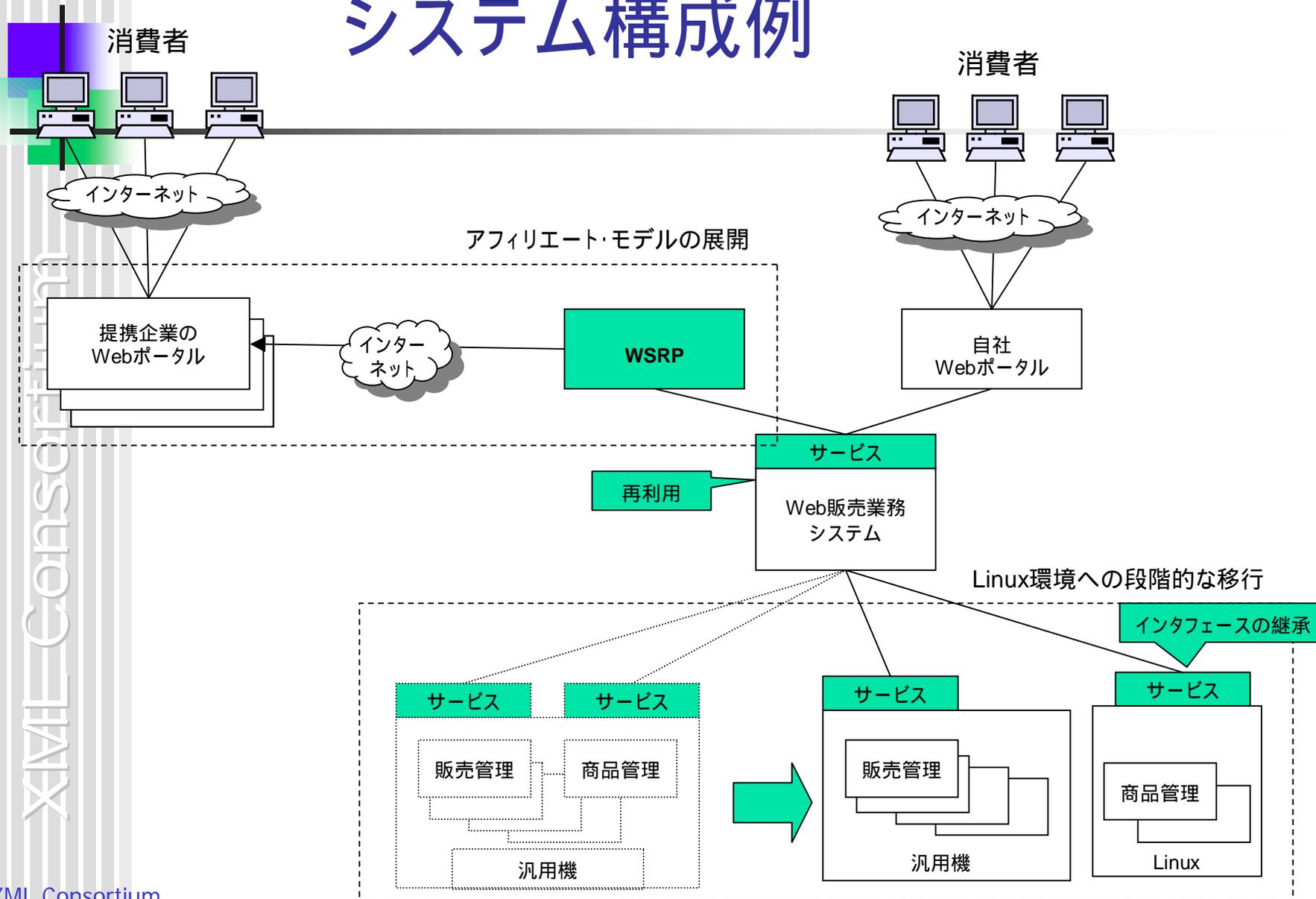
コンポジット、プリミティブ比較

	コンポジット・サービス	プリミティブ・サービス
目的	特定業務の支援 業務間連動の効率化	アプリケーション開発の生産性と品質向上
関心のある対象者	業務担当、事業企画(業務横断的 にな視点を持つ)	情報システム部門
分析機軸	ビジネスプロセス・モデル	ビジネスエンティティ・モデル
サービス統合方向	コンポジット・サービスは水平的に統合 される	プリミティブ・サービスは垂直的に 統合される
機能特性	機能/データ統合 ワークフロー 入出力データの編集・加工	トランザクション処理 バッチ処理 データ参照・更新
実装の関心	サービス・インタラクションと複合 オーケストレーション コレオグラフィ	サービス・ラッピング
設計上の要件	コンポジット・サービス間の依存関係	プリミティブ・サービス間の独立性

部門最適を考慮したシステム構成例

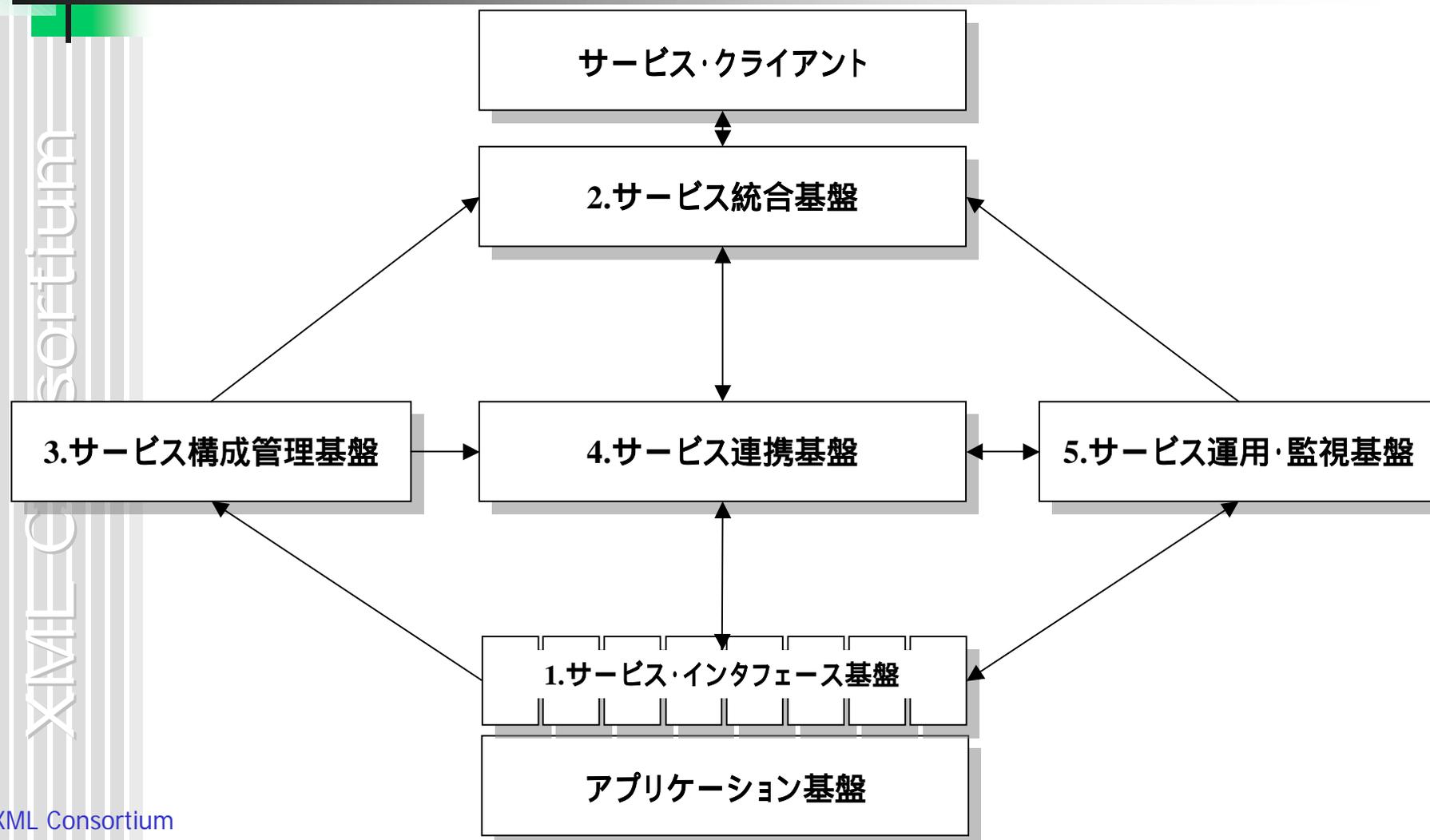


ビジネス変革とIT革新に備える システム構成例

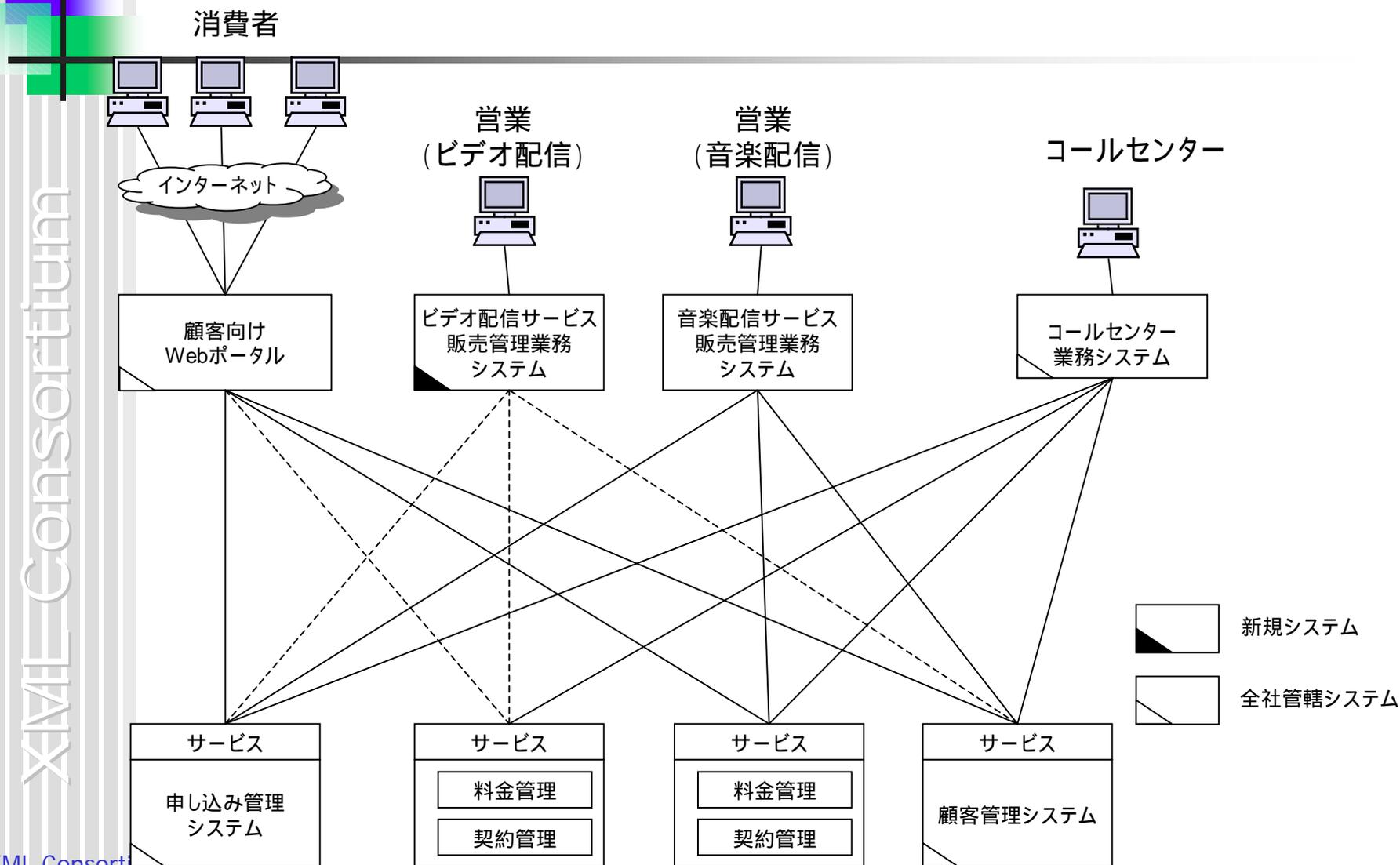


XML Consortium

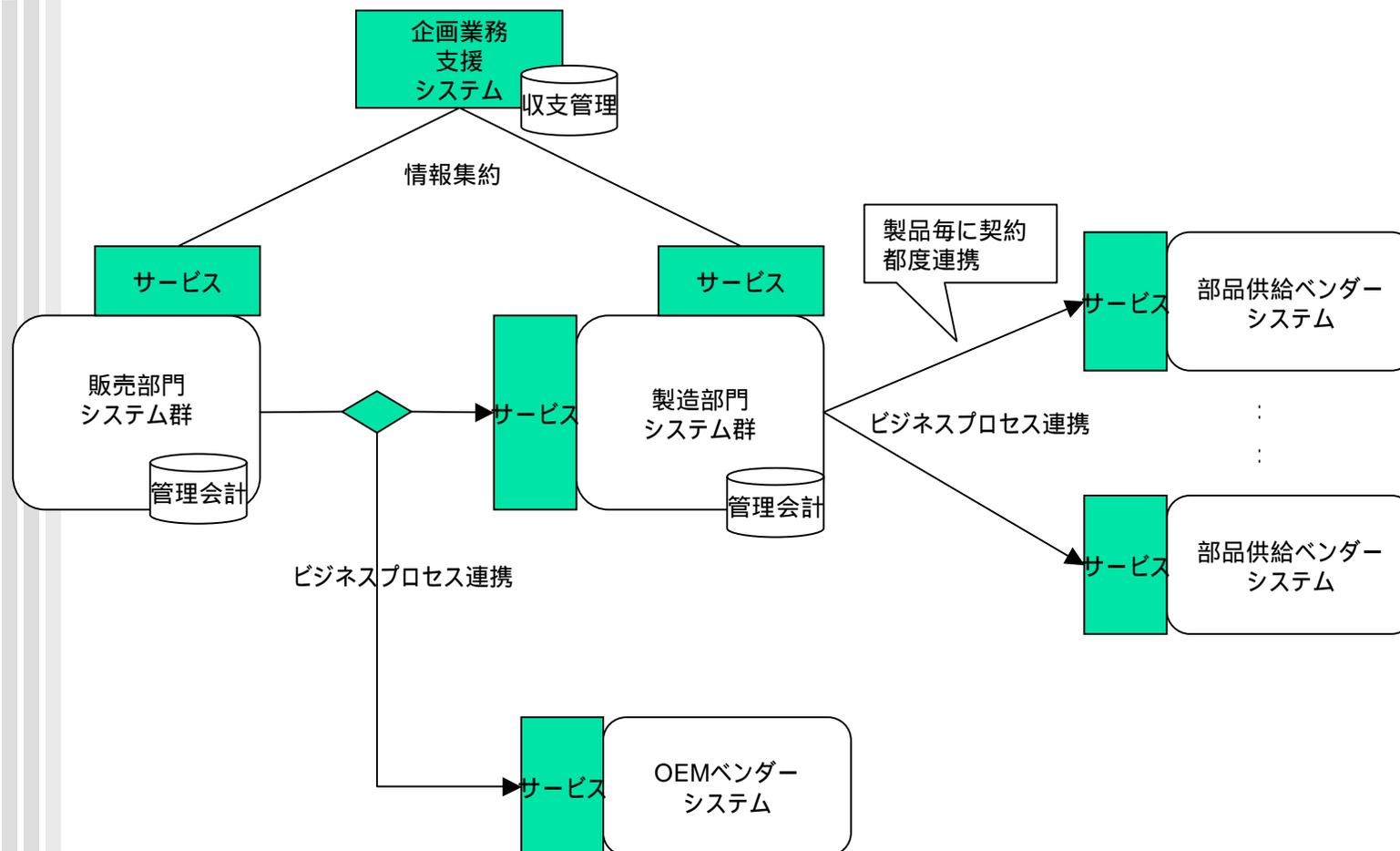
SOA技術アーキテクチャ基盤関連図



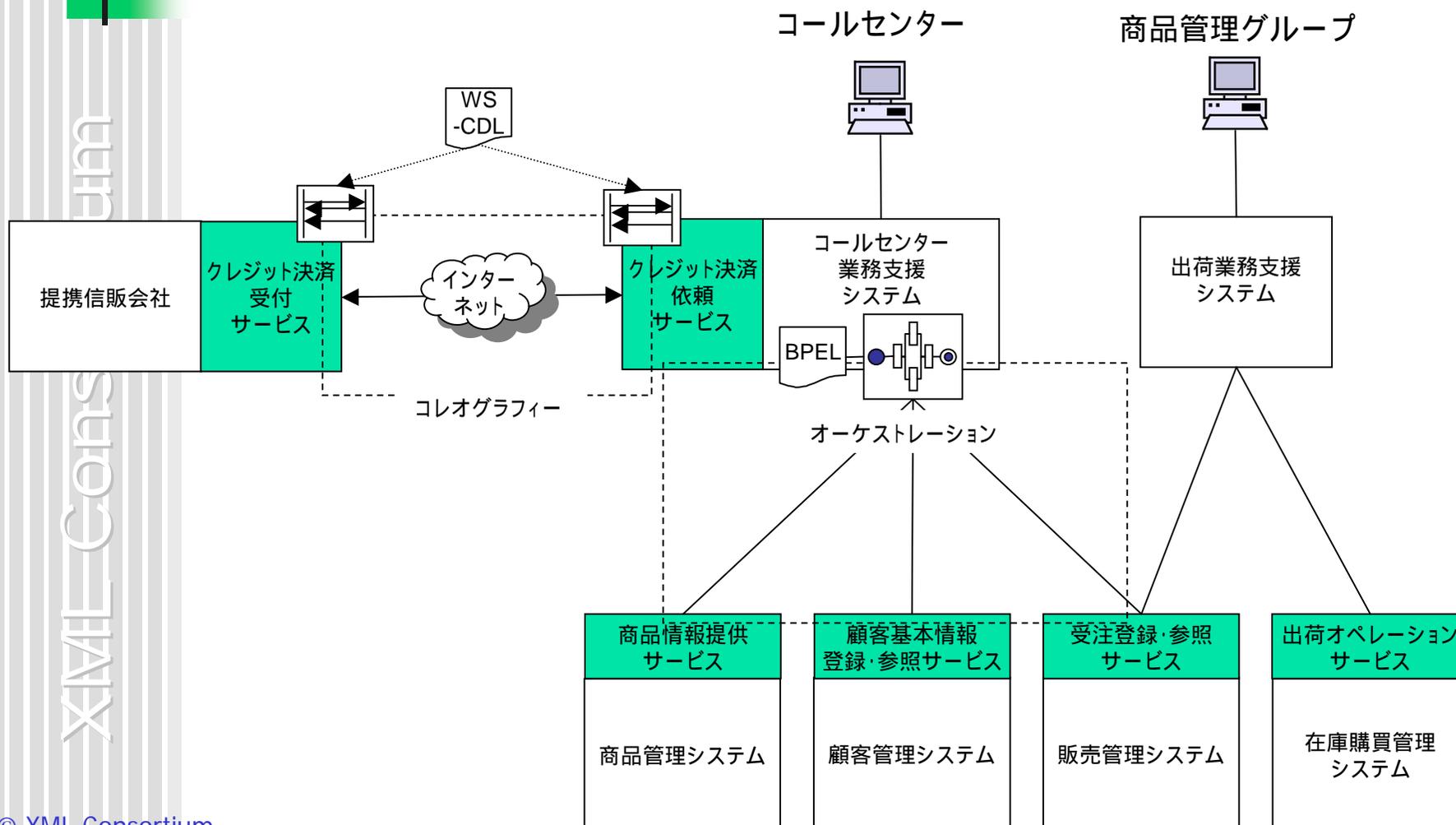
全体最適システム構成例



独立性の強い部門、 企業間のシステム連携例



サービス構成例



1

なぜSOAが必要か？

2

アーキテクチャ概念

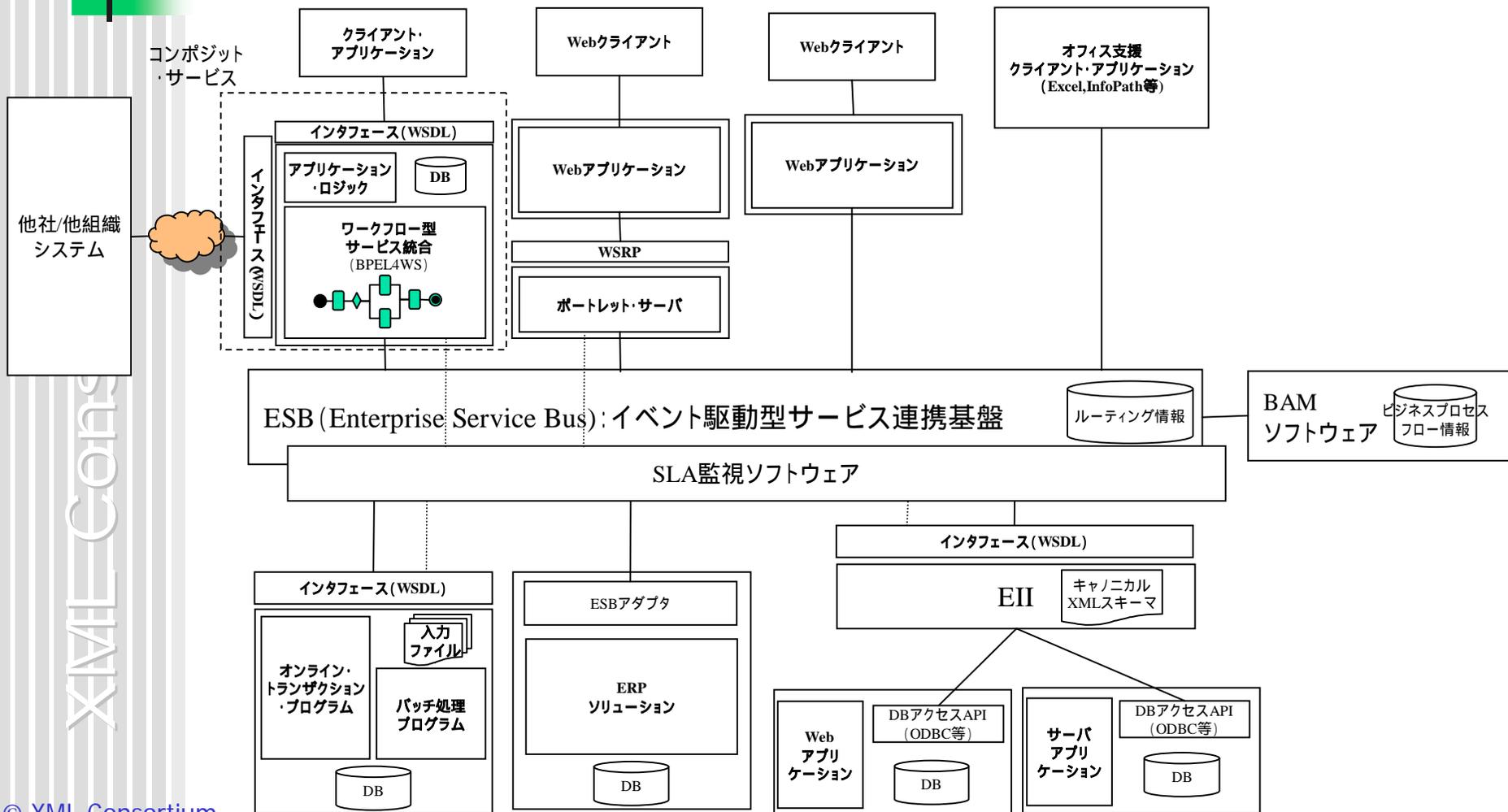
3

技術アーキテクチャ

4

モデリングのポイント

SOAシステム環境



XML Consortium

1

なぜSOAが必要か？

2

アーキテクチャ概念

3

技術アーキテクチャ

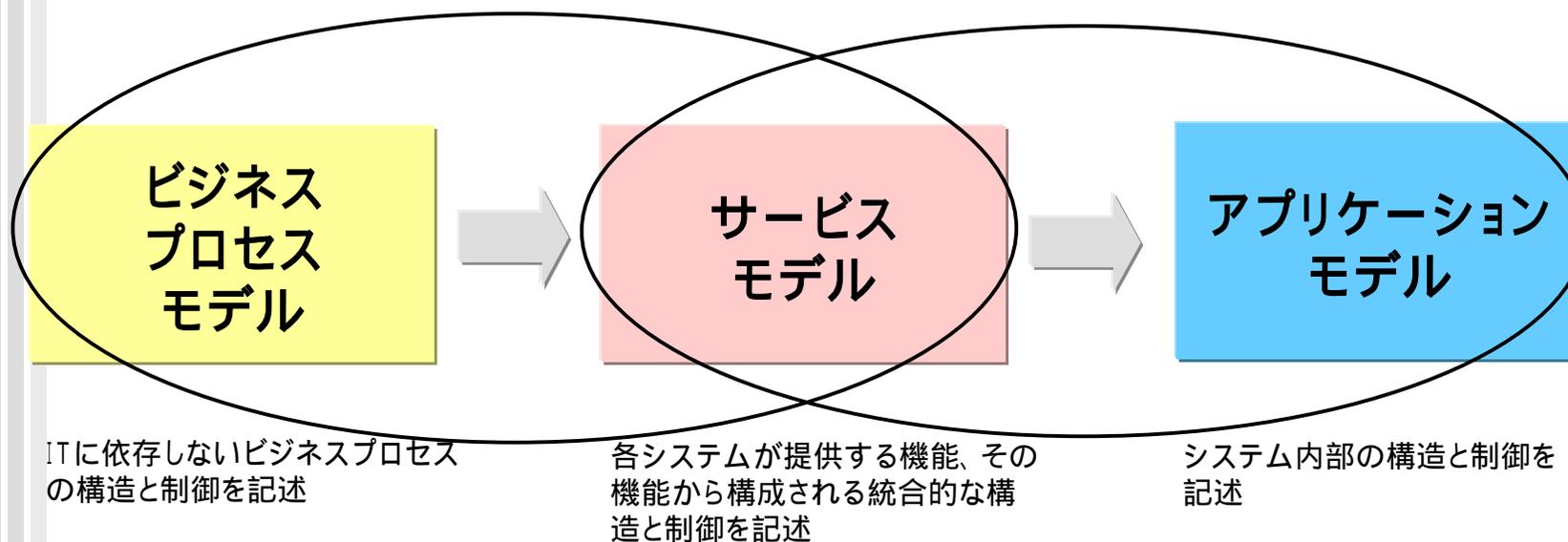
4

モデリングのポイント

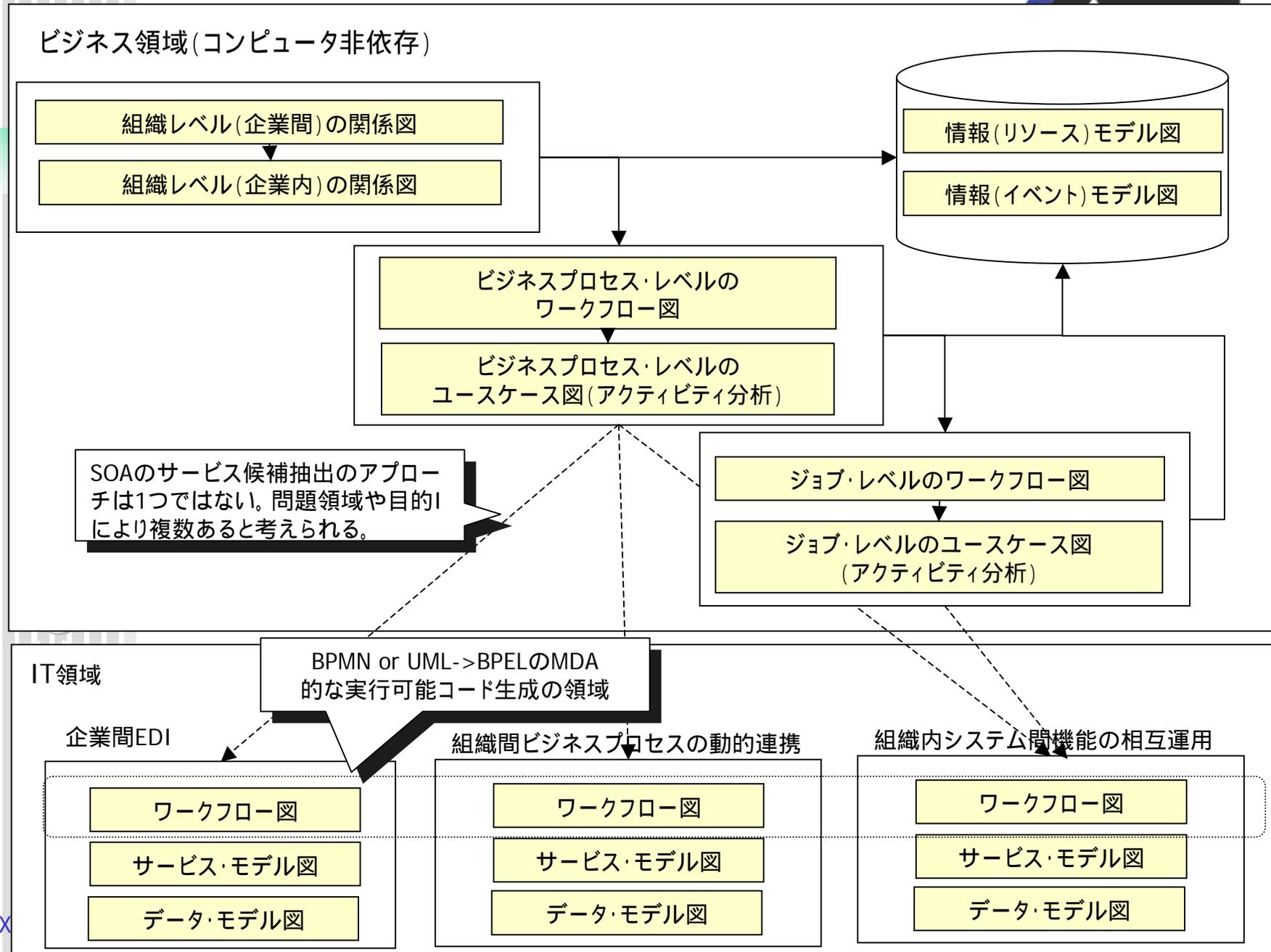
モデリング・プロセス

事業部門スタッフ

情報システム部門スタッフ

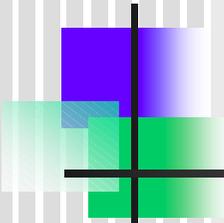


SOAのCIM-PIMモデル関連図



SOA適用全体フレームワーク

	プロセス (Composite Service)	データ (Primitive Service)	全体構造 (Structure)	普及・展開 (Deployment)
分析工程 (Analysis)	ビジネスプロセス・モデルの作成	ビジネス・エンティティの分析	変化対応方針の策定 SOA適用範囲の決定	役割・管理体系の分析
設計工程 (Design)	サービス・モデルの作成	データ・モデルの作成	サービス構造化 (レイヤリング、リレーション)	
実装工程 (Implementation)	サービス・インタフェース定義とデータ構造定義		サービス統合方針 (インテグレーション・モデル)	
	アプリケーション実装			
環境工程 (Environment)	既存システム, パッケージS/W連携方針		基盤アーキテクチャの策定	サービス管理機構の確立

A decorative graphic consisting of a vertical black line intersected by a horizontal black line, with a purple square above the intersection and a green square below it, both with a gradient effect.

まとめ

- モデリング手法、方法論は確立していない
- SOAのアプリケーション像は今後