

SOAとは:

SOAはどこから来て, どこへ向かうのか?

👉 SOAはどこから来たか?

👉 SOAの現在

👉 SOAはどこへ向かうのか?

👉 まとめ

青山 幹雄

南山大学 数理情報学部 情報通信学科

情報処理学会 技術応用運営委員会

mikio.aoyama@nifty.com, <http://www.nise.org>

1. SOAはどこから来たか?



SOAはどこから来たか?

Old Wine in New Bottle?

👉 Webサービスは何も新しい?

👉 CORBAがあったではないか?

👉 同じ味なら「もう飲み飽きた」?

👉 本質的な違いはあるか?

Webサービス
相互運用性,
プラットフォーム独立,
...

Webサービス
CORBA



CORBA, COM
相互運用性,
プラットフォーム独立



SOAはどこから来たか？

分散オブジェクト環境の「失敗」から

分散オブジェクト環境, EAI/MOMが

👉 目標として, 達成できなかったこと

👉 相互運用性, プラットフォーム独立性

➡ 複数標準の対立
(CORBA, COM)

➡ SOC (Service-Oriented Computing): 「サービス」
= 新たなプラットフォーム

👉 新たに起きた環境の変化

👉 インターネット, e-Business

➡ クライアント/サーバ
とインターネットの
アーキテクチャ不適合

➡ SOA (Service-Oriented Architecture):
動的アーキテクチャ

EAI: Enterprise Application Integration, MOM: Message-Oriented Middleware

CORBA: Common Object Request Broker Architecture, COM: Component Object Model

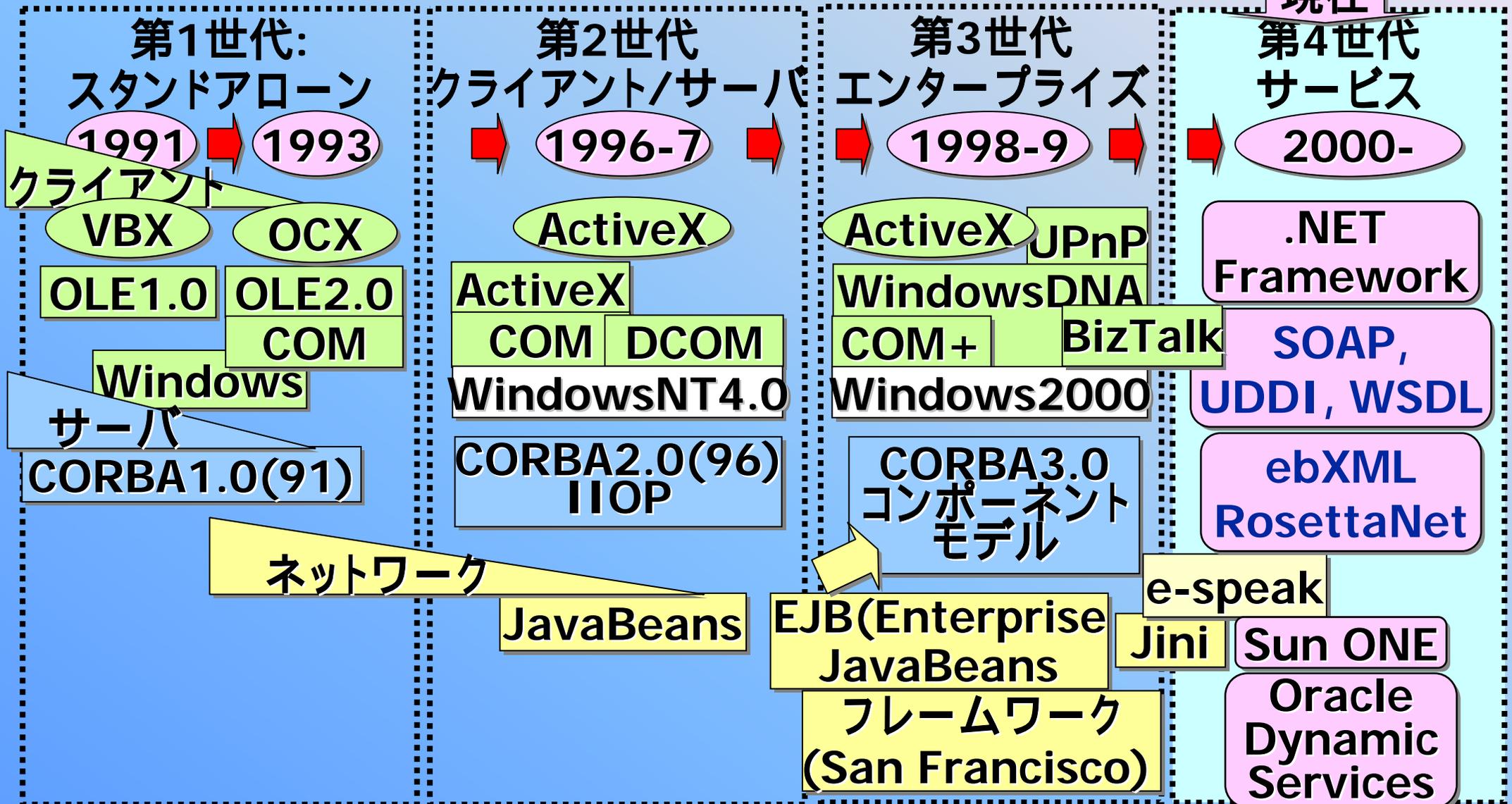
SOAはどこから来たか?

コンポーネントからサービスへ

👉 90年代: コンポーネント技術

2000年代: サービス技術

現在

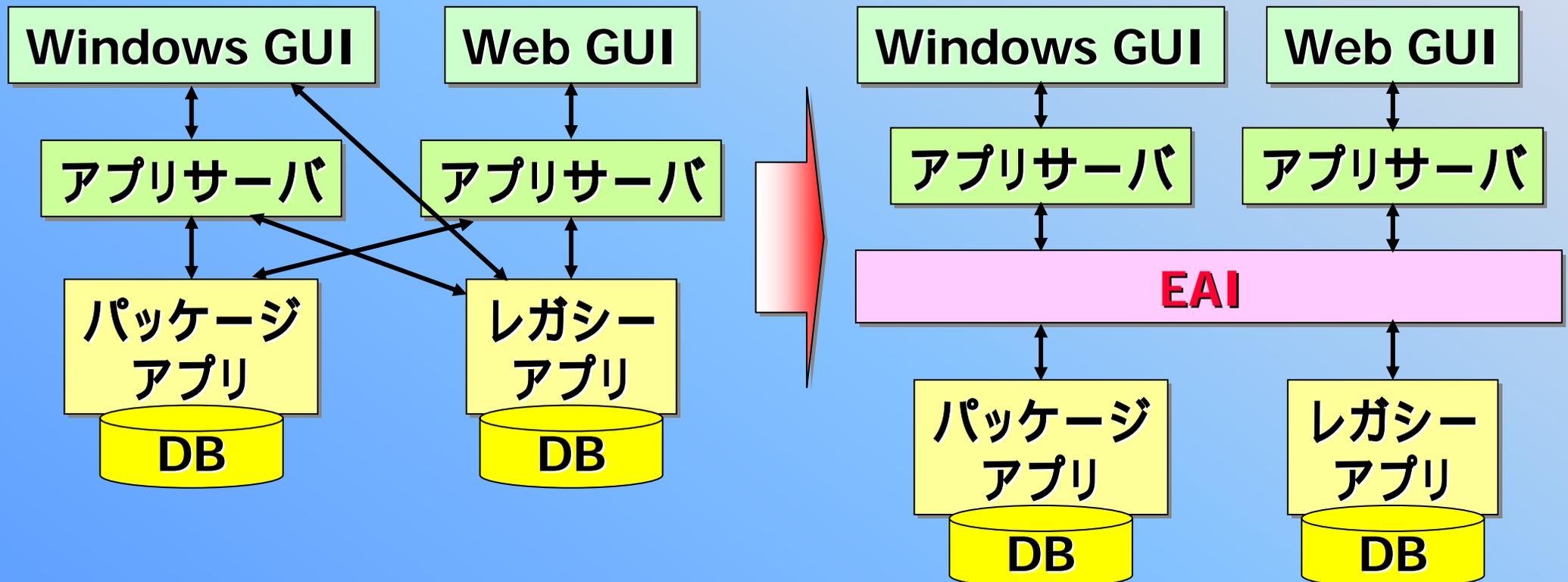


SOAはどこから来たか?

企業システム統合(EAI)のコンセプト

👉 EAI (Enterprise Application Integration)

- 👉 異なるアプリケーションを統合する技術体系とソフトウェアアーキテクチャ
- 👉 アプリケーション単位(疎粒度): ERPなどを中心とする統合

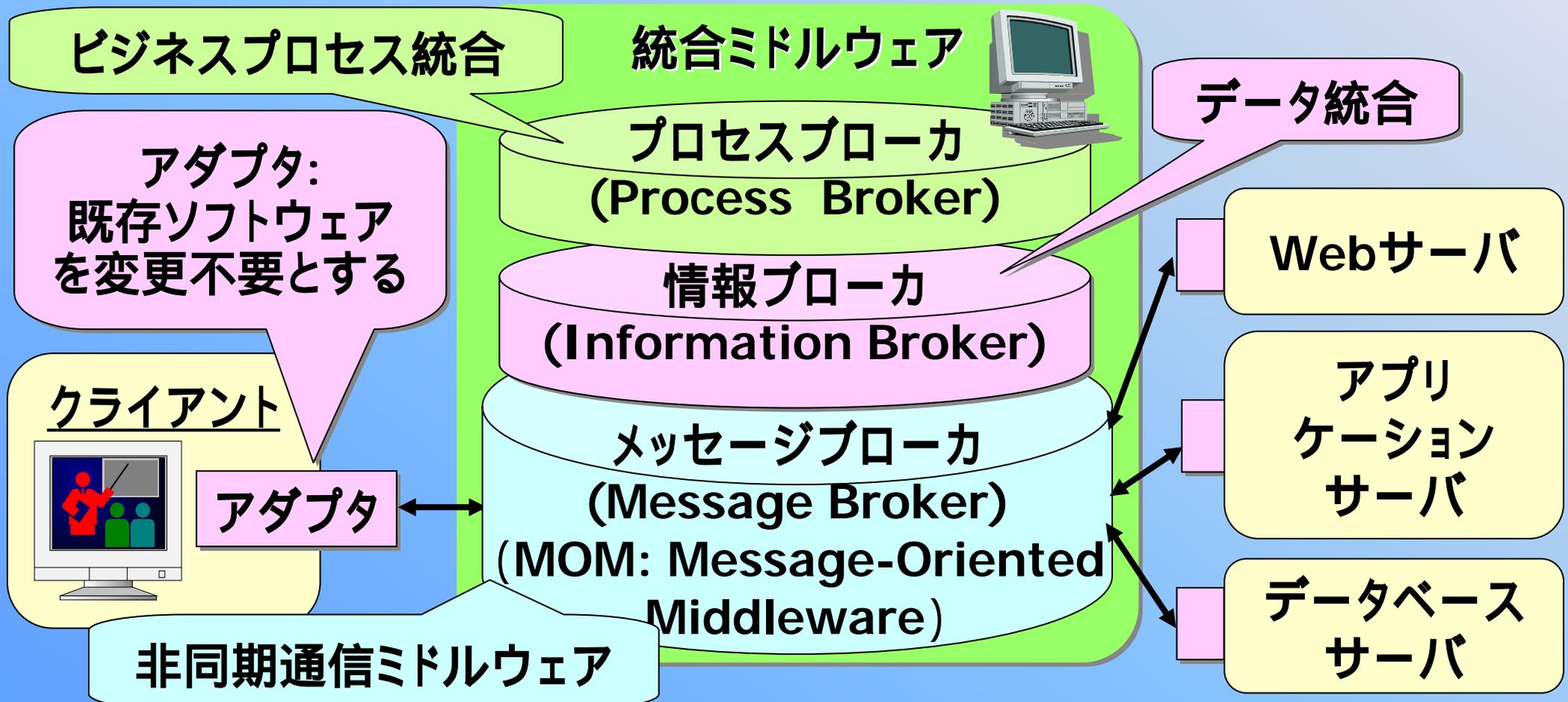


SOAはどこから来たか?

EAIのアプローチ: ブローカーアーキテクチャ

標準統合アーキテクチャの提供

- 👉 メッセージブローカーとインタフェースアダプタの提供
- 👉 ベンダ個別インタフェースの問題: アダプタだらけ



SOAはどこから来たか？

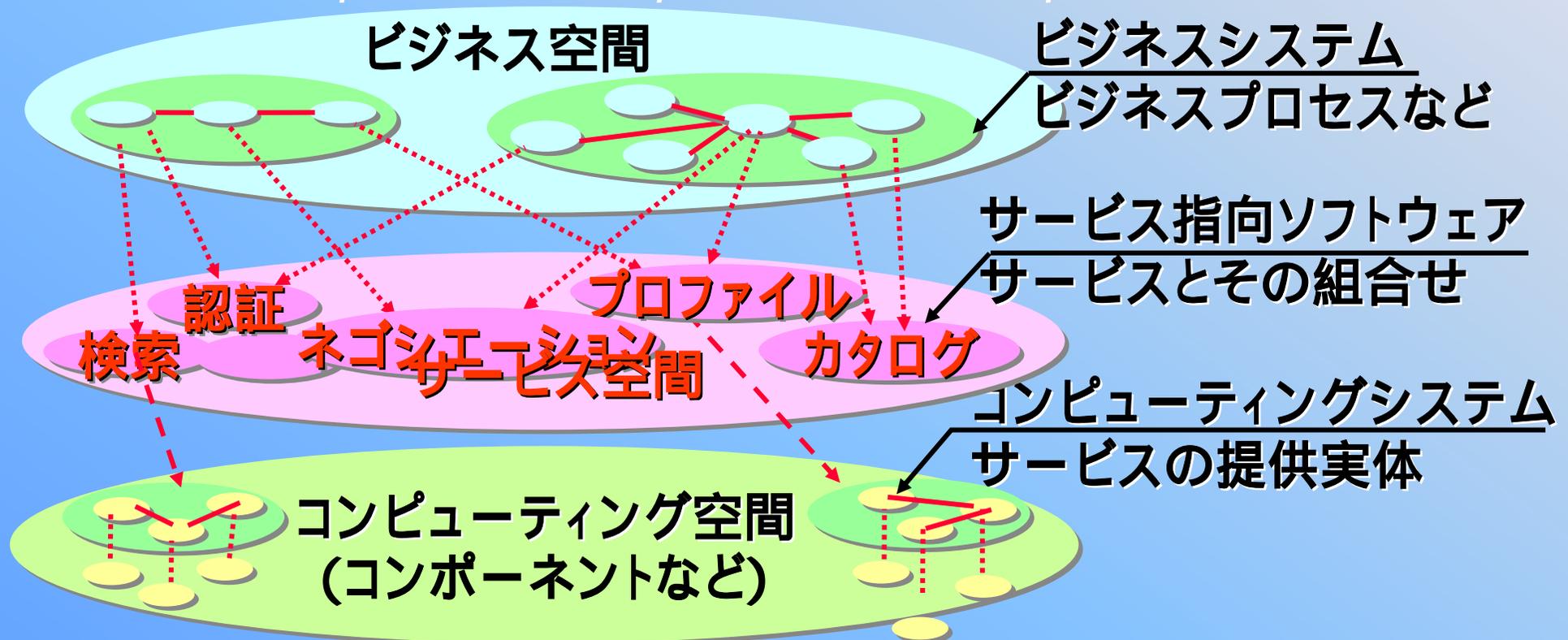
新たなインタフェースモデルの研究

- ☞ CORBA, COMを超えるインタフェースモデルの開発
- ☞ Web上のメタ言語によるインタフェース定義[青山ほか]
 - ☞ 1997年: SCL(Software Commerce Language): HTMLのメタタグを使用したインタフェース定義言語
 - ☞ 1999年: XSCL (eXtensible SCL): XMLを使用したインタフェース定義言語
- ☞ 類似の研究
 - ☞ BML(Bean Markup Language) S. Weeraranawa, IBM T J Watson Research Center <WSDLのリードアーキテクト> ほか

SOAはどこから来たか？

サービス指向ソフトウェア(SOC)とその位置づけ

- ☞ サービス=Web上で共通にご利用可能なソフトウェアの機能
 - ☞ 共通の言語基盤: XMLでインタフェースをラッピング
- ☞ サービス指向ソフトウェア(SOC: Service-Oriented Computing)
 - ☞ Webサービス, ピアサービス, グリッドサービス, モバイルサービス



SOAはどこから来たか？ ビジョンと現実

👉 サービス = 単一共通基盤

👉 IBM, Microsoft, Sun, Oracle,

...

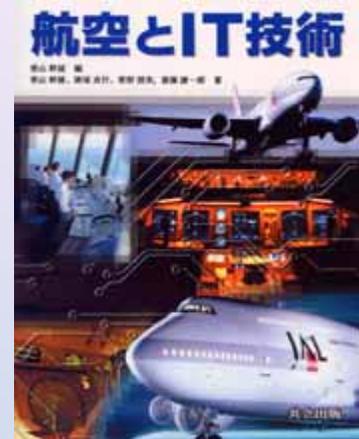


「すべての鍵は
インテグレーションと
インフラストラクチャ
が握る」
IBM CEO(当時)
ガースナー,
e-business
Conference Expo.,
2000年12月」



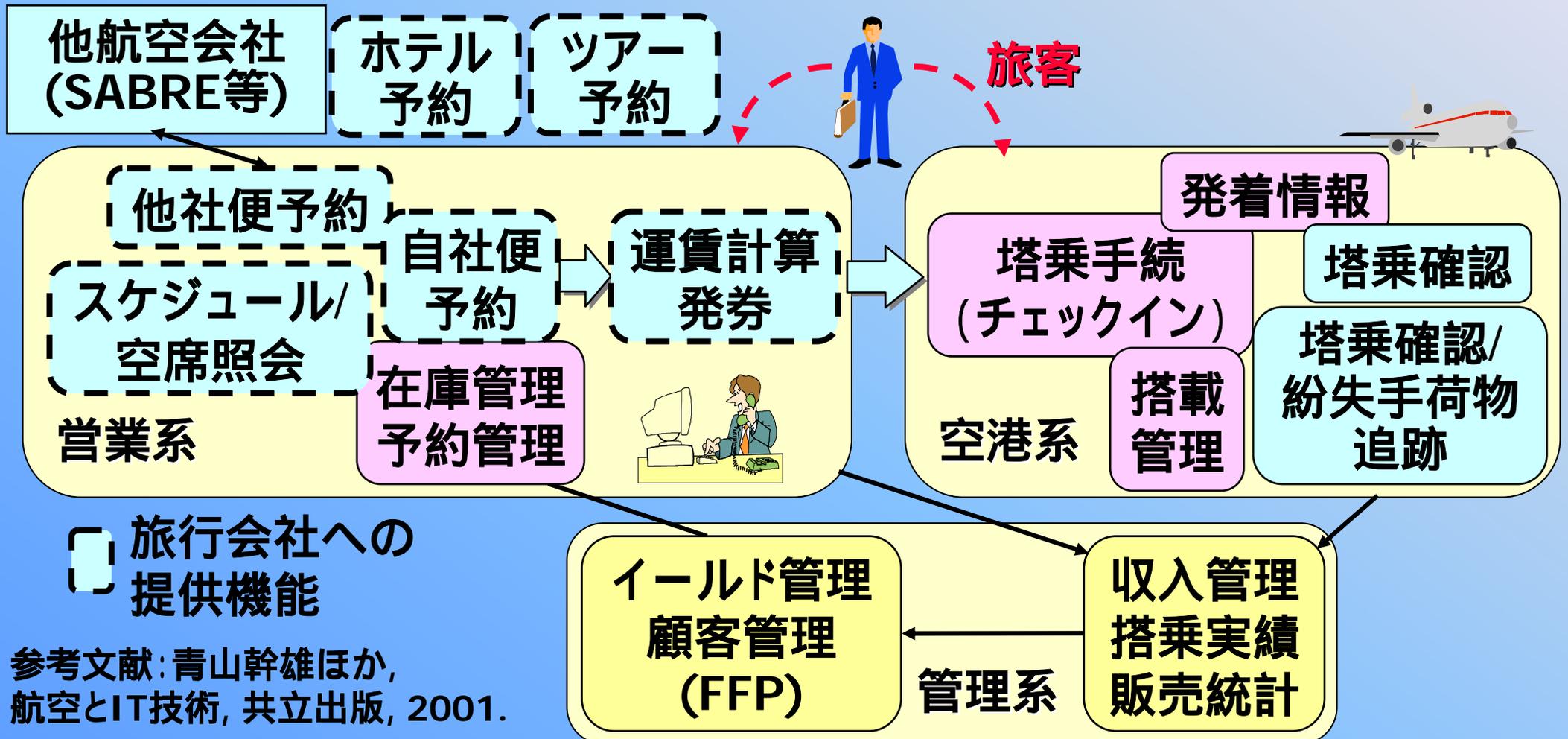
SOAはどこから来たか?

!(統合)が付加価値を生む



✎ 航空旅客業務システム

- ✎ 業務プロセスの統合: 予約から顧客管理まで統合
- ✎ チャネル拡大: 他社, 他旅行業務システム(ホテル, レンタカー等)と統合



参考文献: 青山幹雄ほか,
航空とIT技術, 共立出版, 2001.

SOAはどこから来たか？

ネットワーク = ビジネス

👉 **ネットワーク型ビジネス: ネットワークがコアコンピタンス**

👆 予約業務の効率化から総合予約サービスによる価値増大へ

👋 **SABRE(アメリカン航空系), AXESS(日本航空系)**

👋 **情報システムの進化がビジネスモデル創出: 収益管理, マイレージ**

👆 **成功要因: 経営目標と情報システム開発が一致**

👉 **インターネットによる加速**



CRS: Computer Reservation System

参考文献: 青山 幹雄, 情報技術と航空の共進化: グローバルな航空ITネットワークの形成, 情報処理, Vol. 44, No. 12, Dec. 2003, pp. 1253-1259.

SOAはどこから来たか？ (企業)情報システムの進化

システムモデルの進化

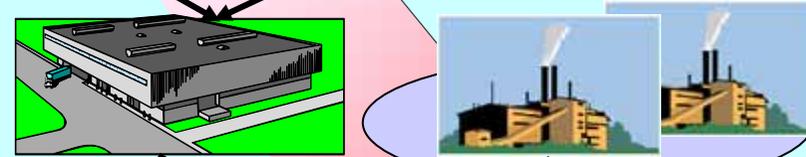
組織内統合(個・点)からネットワークによる組織間連携(社会・面)へ

ストックからフローの制御へ

組織間連携技術の必要性

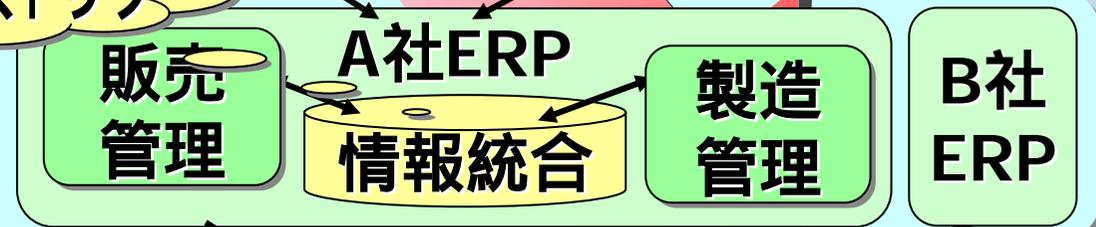


業務システム
例: 販売管理・在庫管理



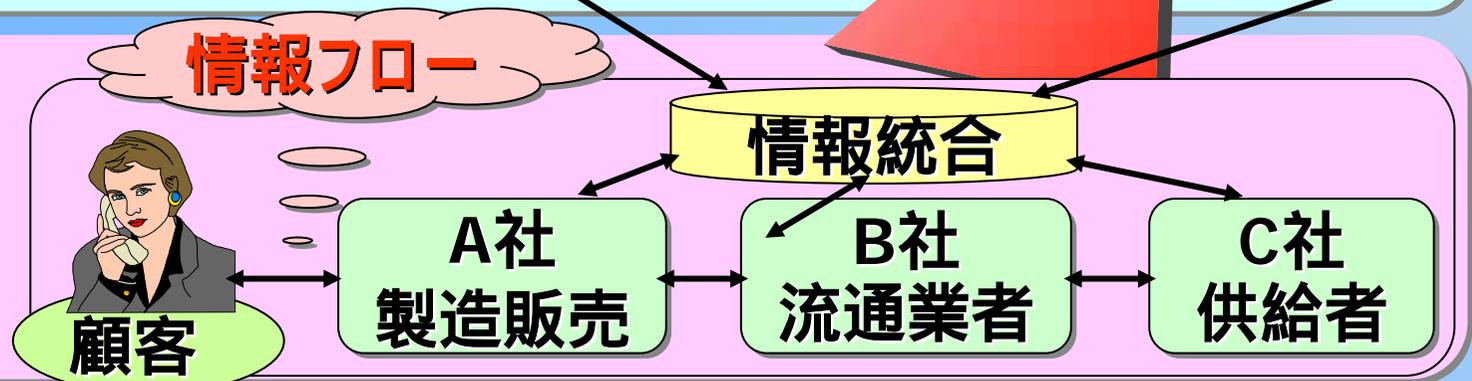
情報ストック

企業システム[80-90年代]
(組織内統合) 例: ERP



ネットワーク
企業システム
(組織間連携)
例: SCM, CRM

情報フロー



SOAはどこから来たか?

統合 = 付加価値を極大化する戦略

👉 組織間連携がもたらす付加価値の増大

- 👉 コスト削減: 組織内では達成できないプロセス全体の最適化
- 👉 付加価値の増大: サービス統合, ビジネスモデル創出, 俊敏性

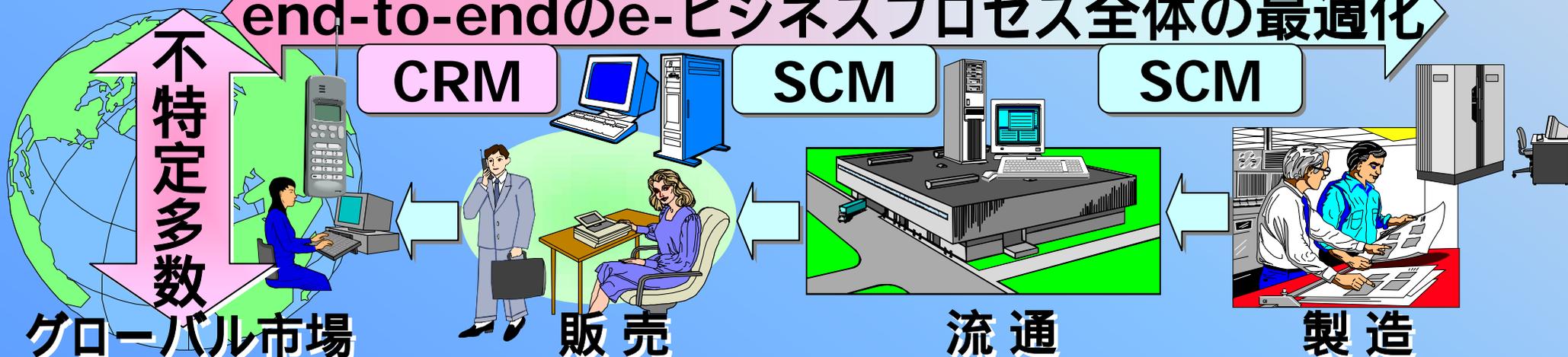
B2C, P2P

- ✍️ 「個」へ直接繋がる
- ✍️ サービスの統合
[販売, 保守, 融資]

B2B, B2M2B

- ✍️ 組織間ビジネス連携・統合
- ✍️ 新たなビジネスモデル
[SCM, eマーケットプレイス]

end-to-endのe-ビジネスプロセス全体の最適化



SOAはどこから来たか？

分散オブジェクト環境の限界

分散オブジェクト環境のアーキテクチャ不適合

👉 クライアント/サーバ: 閉じた・密結合アーキテクチャの問題

👉 クライアント内リポジトリでサーバを特定: 閉じたネットワークが前提

👉 状態結合(State-full)モデル: 密結合

👉 異なる標準(プラットフォーム)間の相互運用性の問題

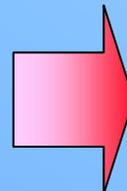
👉 固有インターフェース: 異なる分散オブジェクト環境間の連携が困難

👉 組織間ネットワーク連携の問題

👉 バイナリプロトコル: ファイアウォールを越えた連携が困難



より高い抽象度の共通化
独立性の高いアーキテクチャ



SOA(サービス指向
アーキテクチャ)

SOAはどこから来たか?

分散オブジェクト環境/EAIからWebサービスへ

分散オブジェクト環境, EAIからWebサービスで統一へ

分散オブジェクト環境

1990年頃～

対象: オブジェクト指向アプリケーション開発

スコープ: 単一アプリ(クライアント/サーバ)

基盤技術: オブジェクト指向ミドルウェア/
アプリケーションフレームワーク
(遠隔手続き呼出し)

標準化: 複数標準の並立, 相互運用困難

EAI

1995年頃～

対象: ERPを中心とするアプリケーション統合

スコープ: 企業全体(企業内)

基盤技術: メッセージ指向ミドルウェアと
インタフェースアダプタ

標準化: なし(ベンダ固有)

Webサービス

2000年～

対象: オブジェクトから
アプリケーションまで

スコープ: 企業間

基盤技術: XMLベースの
Web上のメッセージング

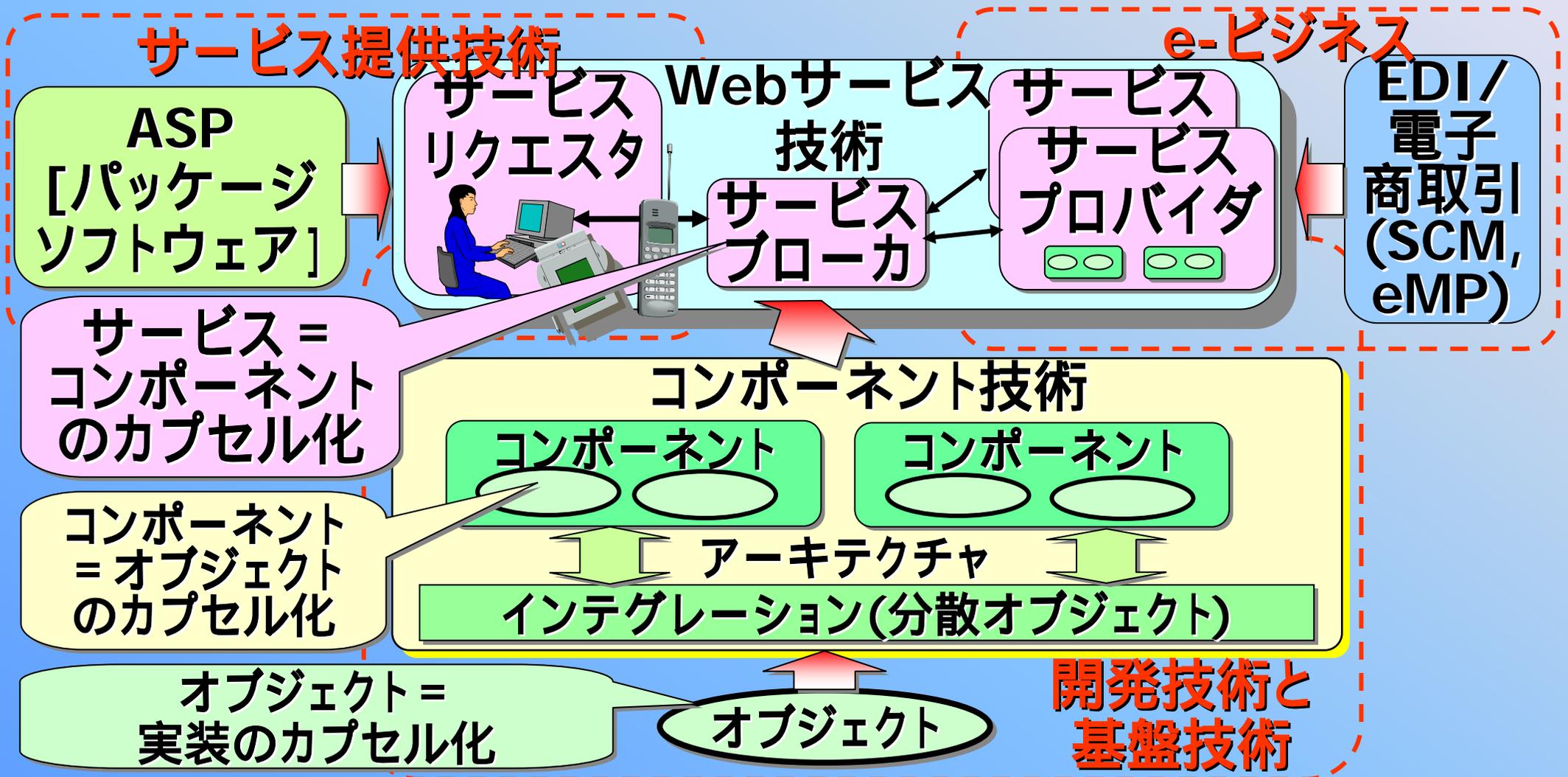
(遠隔手続き呼出し/
メッセージ)

標準化: 基盤技術の
統一 (SOAP, WSDL,
UDDIなど)

SOAはどこから来たか?

オブジェクト・コンポーネント・サービス

- インターネット上でソフトウェアをサービスとして利用
- 進化の3つの推進力: コンポーネント技術, ASP, e-ビジネス



2. SOAの現在



SOAの現在

SOAの本質的特性

👉 SOA=動的アーキテクチャ

- 👉 クライアント/サーバからパブリッシュ/サブスクライブ (Publish/Subscribe)アーキテクチャへ
- 👉 設計時から実行時Lookupへ: コンテンツベース(意味ベース)のサービス発見・接続
- 👉 コンポーネントの所有からサービスの利用へ

👉 ソフトウェア(コンポーネント)の動的結合

- 👉 コンポーネント指向: 部品 of 動的(実行時)組合せ
- 👉 サービス指向: サービス(部品) of 動的な **発見** と組合せ

SOAの現在

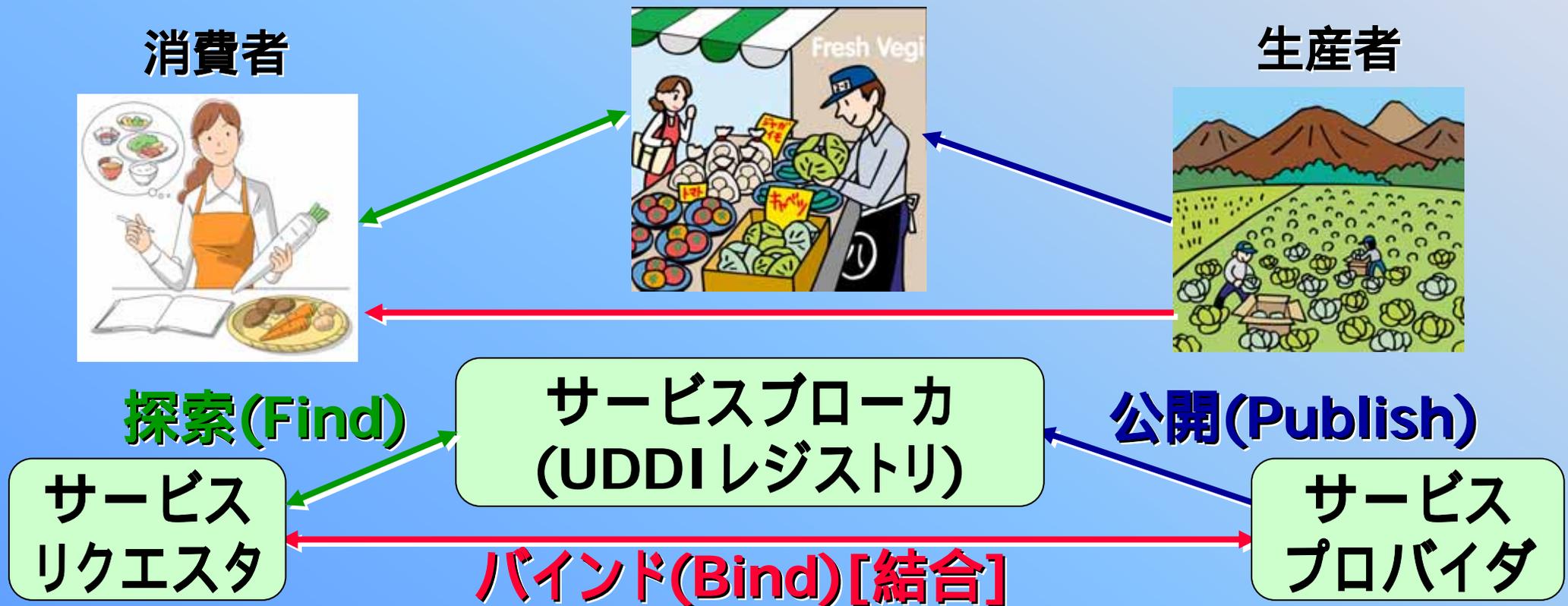
パブリッシュ/サブスクライブアーキテクチャ

パブリッシュ/サブスクライブ(Publish-Find-Bind) パターン

リクエスタ, プロバイダは相互に直接知る必要がない

事前(計画, 設計段階)ではなく, 実行時(買い物する時)に知る

現実社会のビジネスモデルに相似



SOAの現在

パブリッシュ/サブスクライブアーキテクチャ

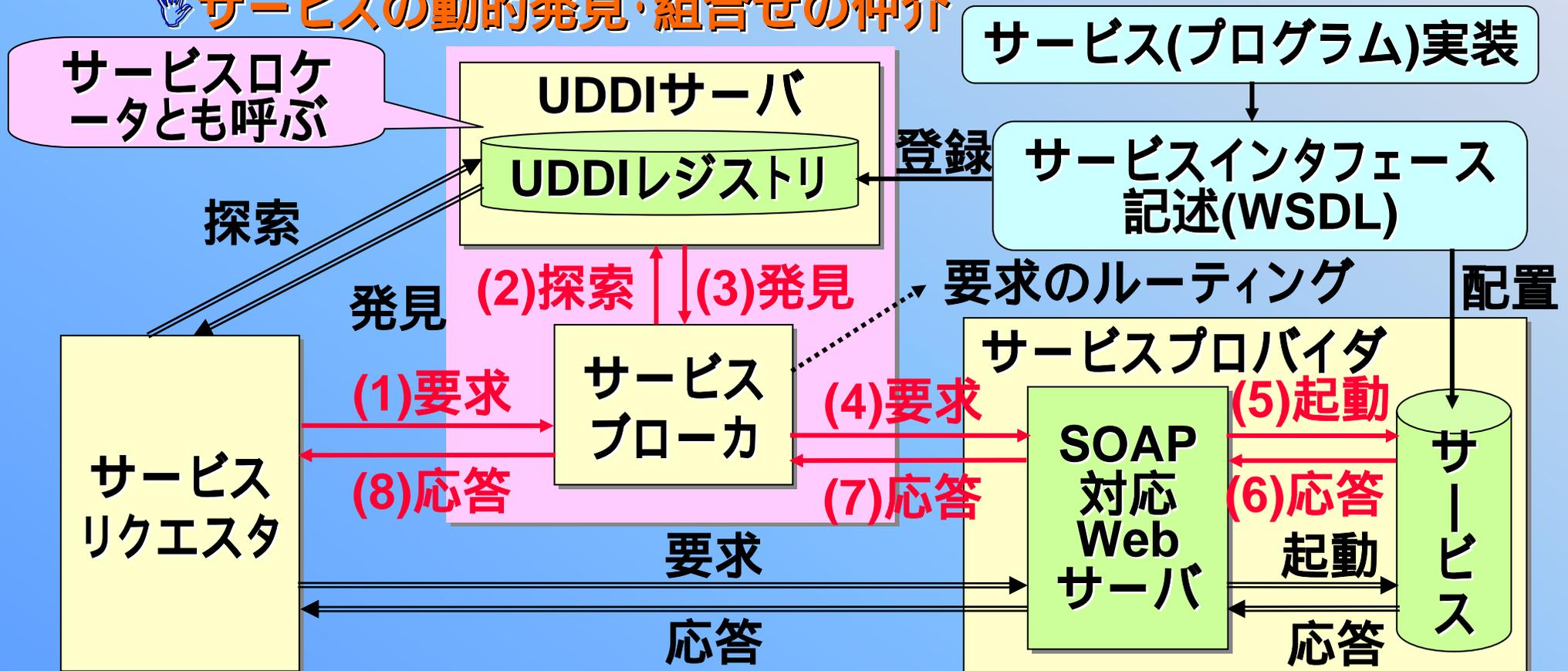
☞ 仲介(サービスブローカ)によるサービス群の動的組合せ

☝ UDDIレジストリ: Publish-Subscribeアーキテクチャ[疎結合]

☞ 出版社と購読者との関係(直接的関係はない)

☝ ブローカ: メディエータアーキテクチャ[結合 = 本屋]

☞ サービスの動的発見・組合せの仲介



SOAの現在

コンテンツ/意味ベースのサービス発見・接続

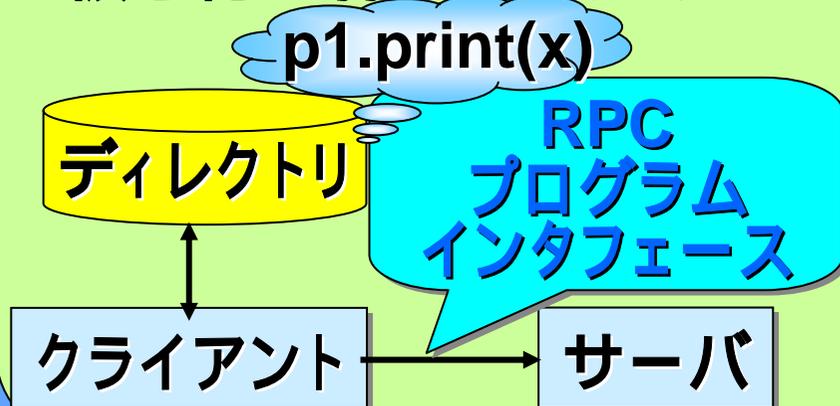
👉 意味ベース(内容に基づく発見と接続)

- 👆 仮想化, 抽象化: 物理アドレスから意味アドレスへ
- 👆 動的な接続: 設計時から実行時に接続の選択

👉 柔軟性と実行時間のトレードオフ

分散オブジェクトアーキテクチャ

- ・クライアント内での直接Lookup
- ・仲介サービスの欠如
- ・仮想化の弱さ: アドレスベース



サービスアーキテクチャ

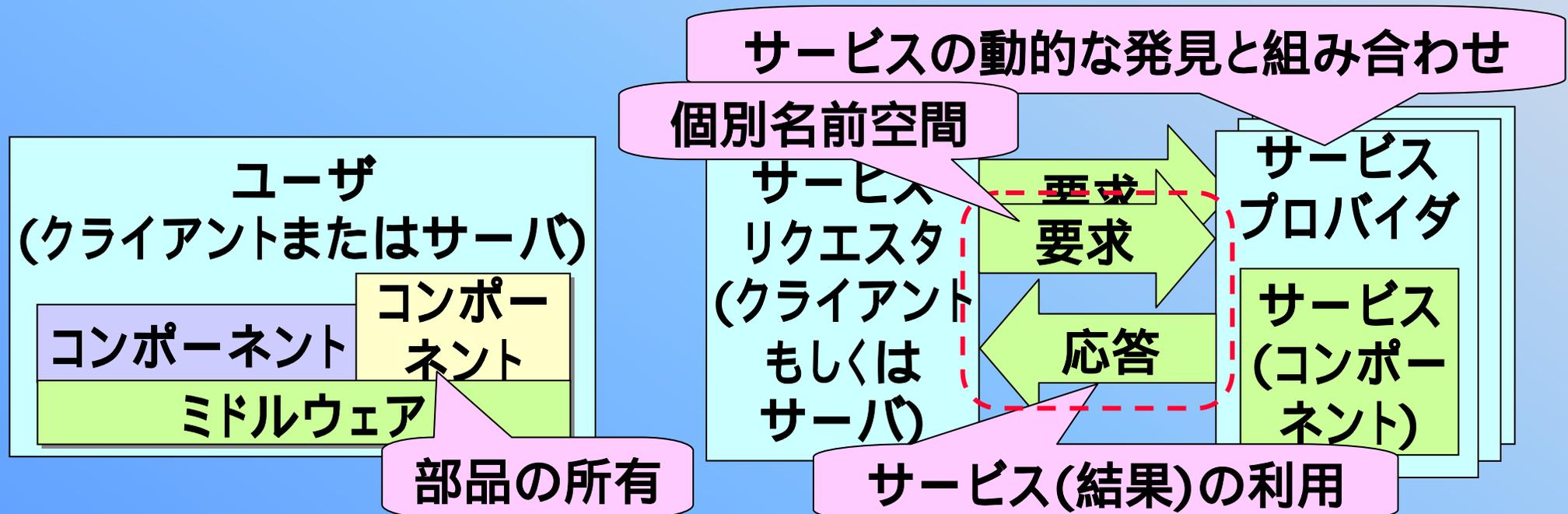
- ・属性ベースのLookup: 意味定義
- ・個別名前空間
- ・サービス仲介: 問合せ + コンテンツベース
- ・統一したサービスモデル



SOAの現在

コンポーネントの所有からサービスの利用へ

- 👉 所有から利用へ: Web上での分散統合=連邦型サービスアーキテクチャ(Federated Service Architecture)
 - 👉 ユーザにおける部品のご合せからプロバイダにおけるサービス(結果)の利用とご合せへ
 - 👉 個別名前空間: 特定の相手間で意味定義を共有

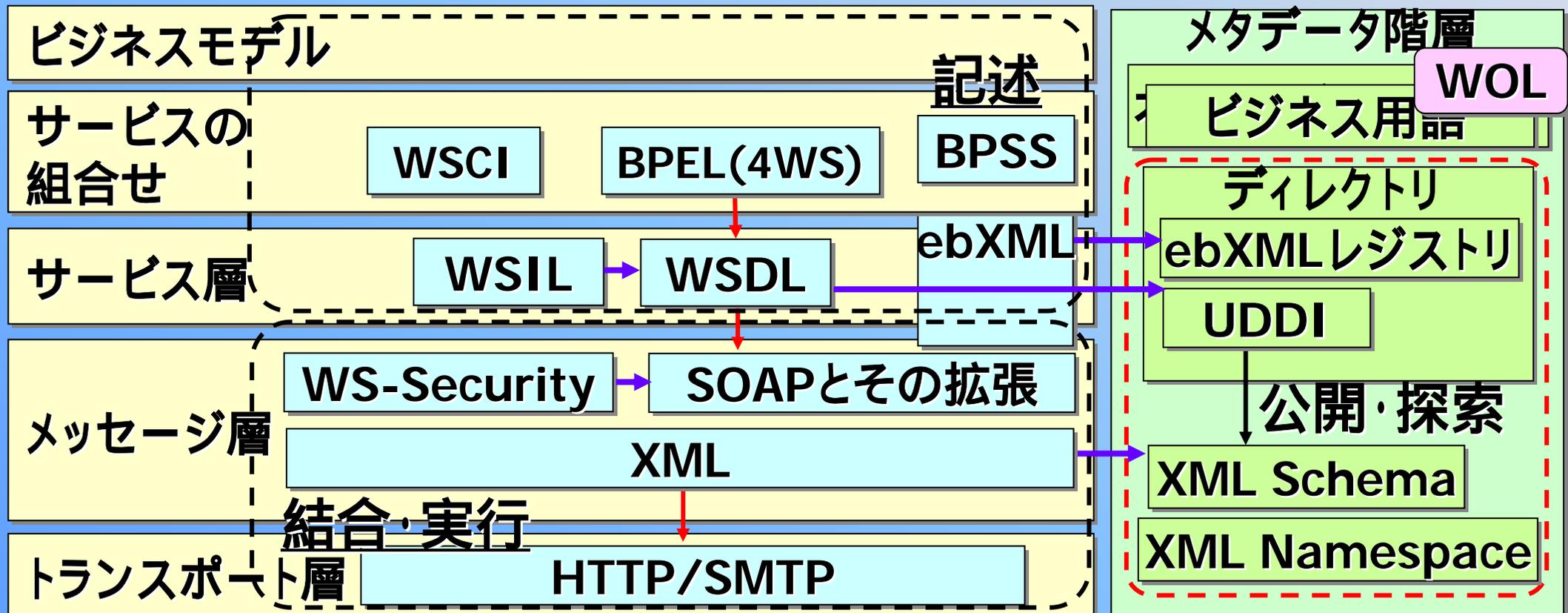


SOAの現在

主要基盤技術の成熟とビジネスプロセスでの競争

☞ XMLベースのソフトウェア技術の階層的体系化

- ☞ サービスの記述, 公開・検索, メッセージング, 結合・実行
- ☞ 情報の意味表現の利用
- ☞ 基盤技術(SOAP, WSDL, UDDI)は安定化
- ☞ ビジネスプロセスによるサービスの組み合わせ, セキュリティ等が課題



SOAの現在

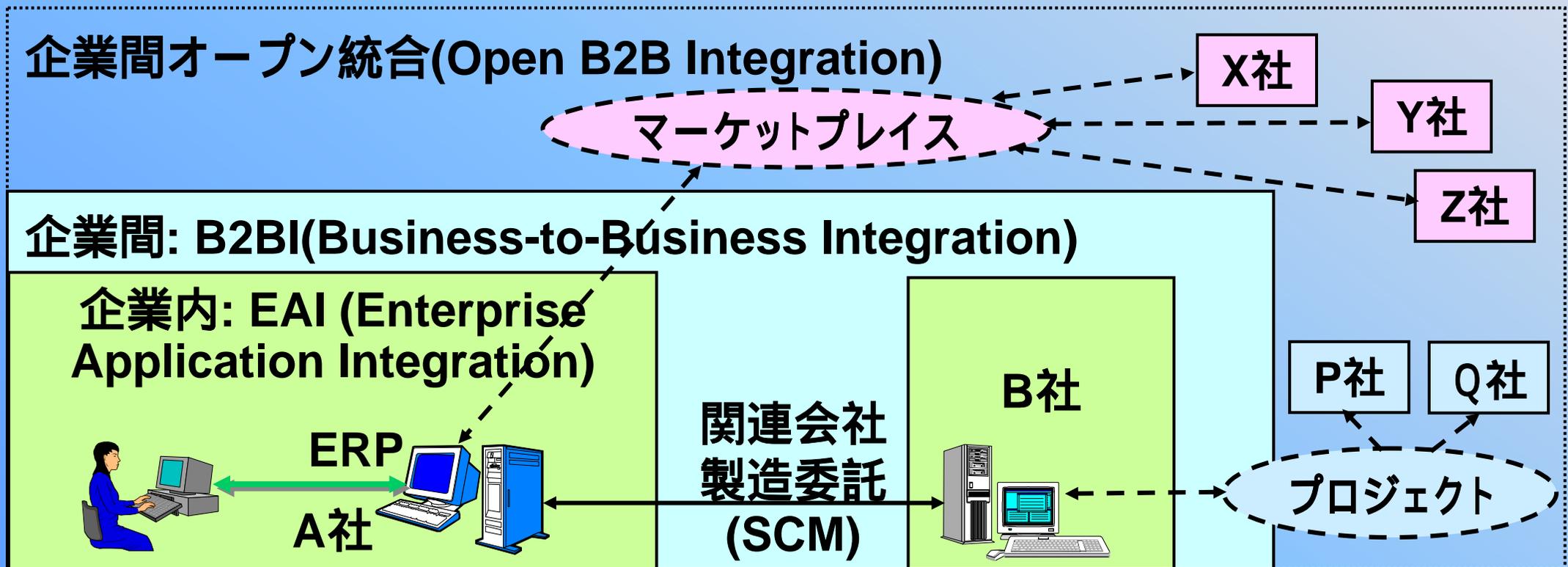
Webサービスによる統合のビジネスパターン

👉 Webサービスによる統合の3段階

- 👉 企業内: EAI 特定企業間(B2BI) 不特定企業間
- 👉 異なるアーキテクチャ, プラットフォーム, ビジネスプロセス

👉 統合範囲: EAI ~ B2BI

- 👉 統合範囲の決定: セキュリティ, ビジネス慣行



SOAの現在

Webサービスによるシステム統合事例

👉 適用実績は増大しつつある

👉 対象: EAI (inB)とB2BI, SOAP連携

👋 形態: プロセス統合, サービス統合

👋 効果: オープン性(クロスプラットフォーム)

👋 取引範囲/市場の拡大, 標準技術による投資効果向上/期間短縮

👉 対象: リッチクライアント

👋 形態: クライアント/サービス間ソフトウェア連携(SOAP)

👋 効果: クライアントの付加価値増大と独立性/柔軟性

👉 課題

👉 セキュリティなどのリスク

👉 Webサービスによるインテグレーションのあり方(3段階)

👋 オープンマーケット型統合のビジネスリスク

参考文献: 情報サービス産業協会Webサービス調査委員会, 平成15年度Webサービス技術を用いたE-ビジネスアプリケーションの事例と課題の調査研究報告書,

<http://www.jisa.or.jp/committee/websrv2/>

SOAの現在

Webサービスの導入パターンとシステム統合

👉 Webサービスに適したビジネスとアプリケーションモデル

👉 リスクと効果の見極め: EAI主体



パターン	提供者	利用者	効果	難易度	事例
EAI	自社	自社	👉 社内のインテグレーションコストを削減	低	Telenorほか
B2Bサービス提供	自社	他社	👉 自社のコアビジネスをサービスとして社外に公開し新たなビジネスを創出	高	Storebrand Amazon, Google
B2Cサービス提供	自社	個人	👉 自社のコアビジネスをサービスとして社外に公開し新たなビジネスを創出	中	NASDAQ イースト社
B2Bサービス利用	他社	自社	👉 アウトソース利用により社内ITリソースおよび人員にかかるITコスト削減	中	Life Time Fitness

SOAの現在

(リッチ)クライアント主導統合: NASDAQ

👉 NASDAQ Investor's Assistant

- 👉 NASDAQ, Microsoft, PricewaterhouseCoopersの共同プロジェクト(<http://www.nasdaq.com>)
- 👉 NASDAQの財務データ(21社)をWebサービス(SOAP)としてクライアントのExcelから利用して分析などが可能
- 👉 財務データの表現: XBRL (eXtensible Business Reporting Language)
 - 👉 米国公認会計士協会が中心となって定めた, XMLを用いて財務諸表を表現する言語(<http://www.xbrl-jp.org/>)



SOAの現在 (リッチ)クライアント主導統合: NASDAQ

☞ NASDAQ Investor's Assistantの利用例

The screenshot displays the 'Excel Investor's Assistant' interface within a Microsoft Excel environment. The interface is divided into several sections:

- Left Panel (Excel Investor's Assistant):** Contains input fields for 'Fiscal Period' (Quarter: 1, Year: 1997), 'Start' (4, 1997), and 'End' (4, 2002). It also includes an 'Industry' dropdown set to 'All Industries' and a 'Company' dropdown set to 'MICROSOFT CORP'. A table lists symbols and company names, with 'MSFT MICROSOFT CORP' selected. A 'Build Analyses' button is visible.
- Top Right Panel (Analysis Type):** Offers options for 'Single Company, Multiple Measures' (selected) and 'Multiple Companies, Single Measure'. It also includes 'Comparison Type and Date Range' (Quarter, 1997-2002) and 'Companies' (MSFT: MICROSOFT CORP).
- Middle Panels (F/S Measures and Ratios):** Two columns of checkboxes for selecting financial metrics. Under 'F/S Measures', 'Total Revenue', 'Net Income (Earnings)', 'R&D Expense', 'Total Operating Expenses', 'Gross Profit Margin', 'EBIT', 'Current Assets', 'Current Liabilities', 'Inventory', 'Total Assets', 'Total Liabilities', 'Stockholder's Equity', and 'Accounts Receivable (Net)' are checked. Under 'Ratios', 'Operating Margin', 'Net Margin', 'Operating Exp / Revenue', 'Gross Margin', 'R&D Expense / Revenue', 'SG&A Expense / Revenue', 'Basic Net Income / Share', 'Diluted Net Income / Share', 'Receivables Turnover', 'Inventory Turnover', 'Total Debt / Equity', 'Long-term Debt / Equity', and 'ROI' are checked.
- Right Panel (MICROSOFT CORP Chart):** A combined bar and line chart showing financial data from 1997 to 2001. The Y-axis represents values in thousands, ranging from 0 to 30,000,000. The legend includes:
 - Total Revenue (Yellow bars)
 - Total Operating Expenses (Blue bars)
 - Gross Profit Margin (Green bars)
 - Operating Margin (Black line)
 - Net Margin (Purple line)
 - ROI (Green line)

SOAの現在

(リッチ)クライアント主導統合: MS Office 2003

☞ マイクロソフト Office 2003 Research Services

☞ OfficeをWebサービスクライアントとする

☞ Word, Excel, PowerPointなどからWebサービスを起動可能(例: Word文書の文字列のクリックで起動)

☞ Office 2003 Research Services: 翻訳, ほか

☞ サードベンダWebサービス: Amazon Web Service, Google Web API Services, ほか

☞ 従来のコンポーネント(DLL)による拡張との違い: クライアント側の変更不要(Smart Tagを使用しない場合)



参考文献: C. Kunicki, Build Your Own Research Library with Office 2003 and the Google Web Service API, Mar. 2003, <http://www.msdn.microsoft.com/library/default.asp?url=/library/en-us/dnofftalk/html/office03062003.asp>

SOAの現在

意味に基づくサービスブローカの動的仲介

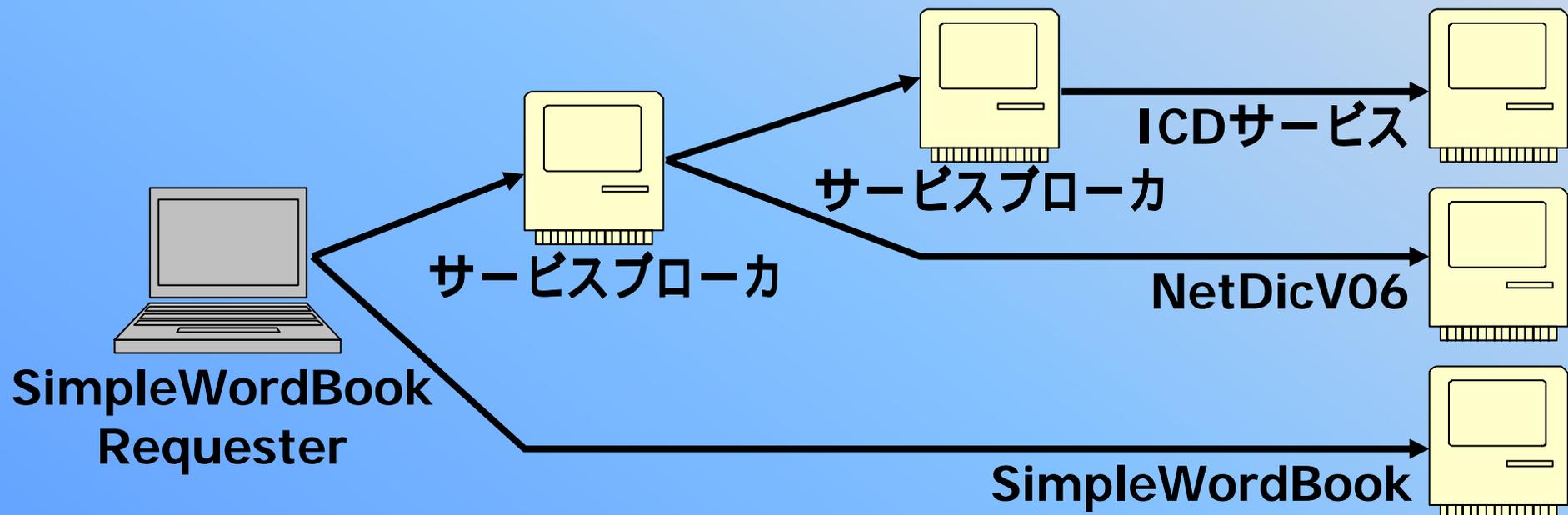
👉 単語帳システム

👉 3種類の単語帳Webサービスをブローカで連携

👉 ICDサービス(@IT): IT分野の辞書

👉 NetDicV06サービス(三省堂): 辞書一般

👉 SimpleWordBook(研究室で開発): ソフトウェア工学の辞書



参考文献: 中村 一仁, 柘植 亮人, 青山 幹雄, 価値を用いたWebサービスの動的連携ブローカとその評価, 情報処理学会ソフトウェア工学研究会, Vol. 2003-SE-144, 2004年3月, pp. 123-130.

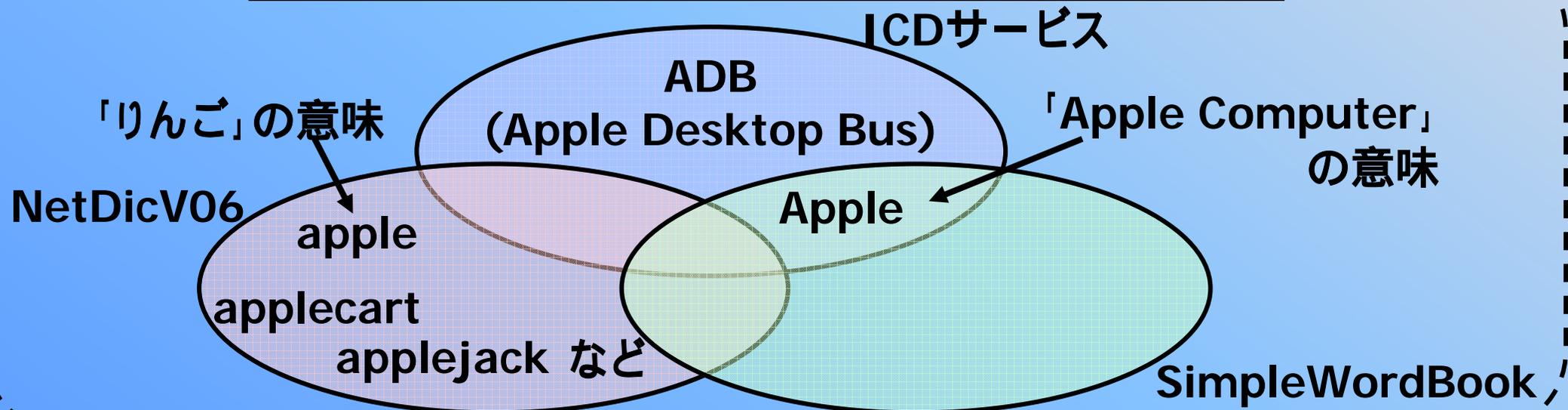
SOAの現在

意味に基づくサービスブローカの動的仲介

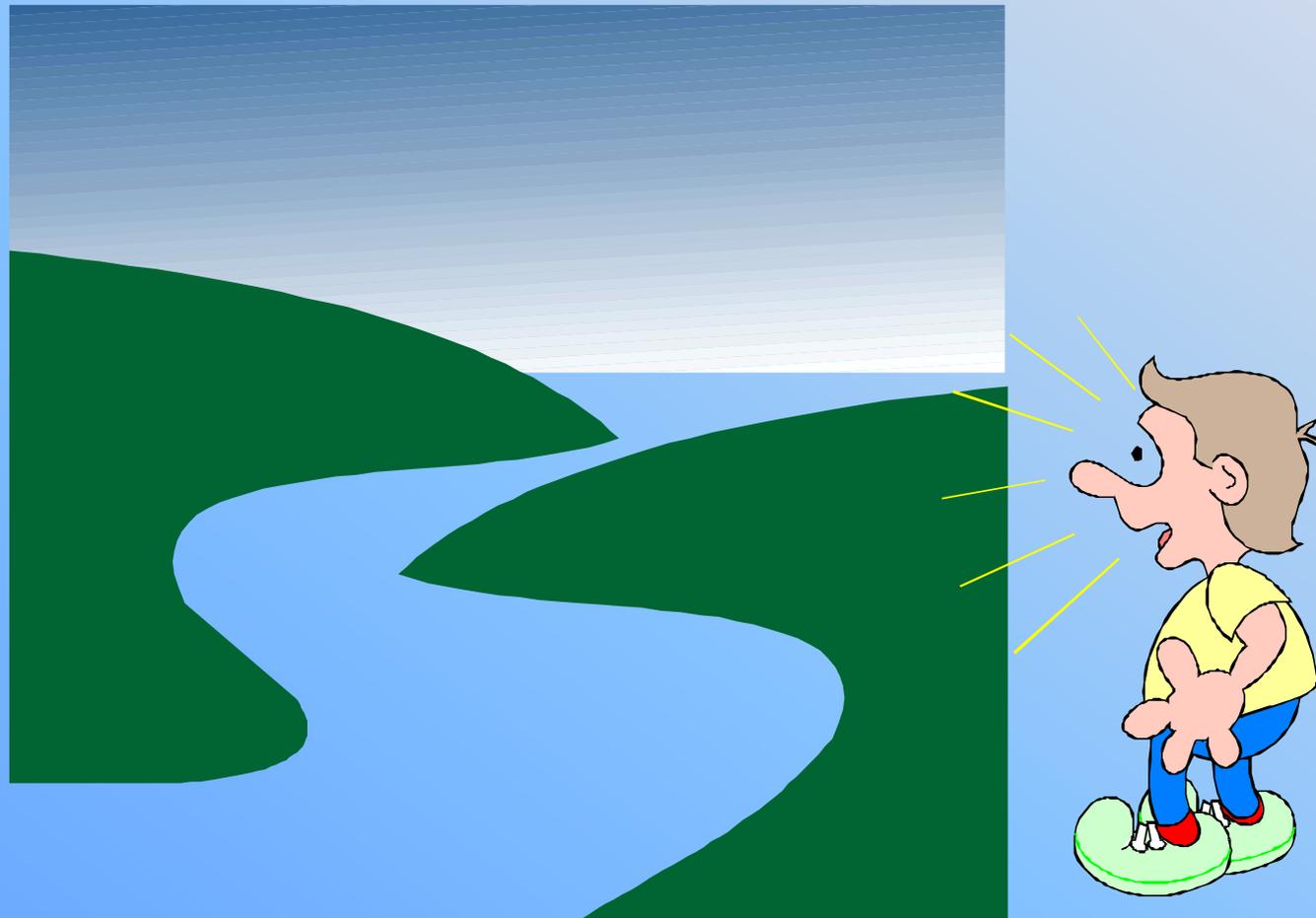
👉 意味の信頼度の向上

- 👉 サービス変換の利用: 意味情報が豊富に, 複数辞書で正しさの確認
- 👉 ICDサービスとSimpleWordBook: 専門用語("Apple Computer"の意味)の意味検索可能
- 👉 ICDサービス: 歴史などの詳しい情報を取得可能
- 👉 NetDicV06サービス: "Apple"を拡張した単語・慣用句の検索可能

例: "apple"を検索語とした部分一致検索の結果



3. SOAはどこへ向かうのか？



SOAはどこへ向かうのか?

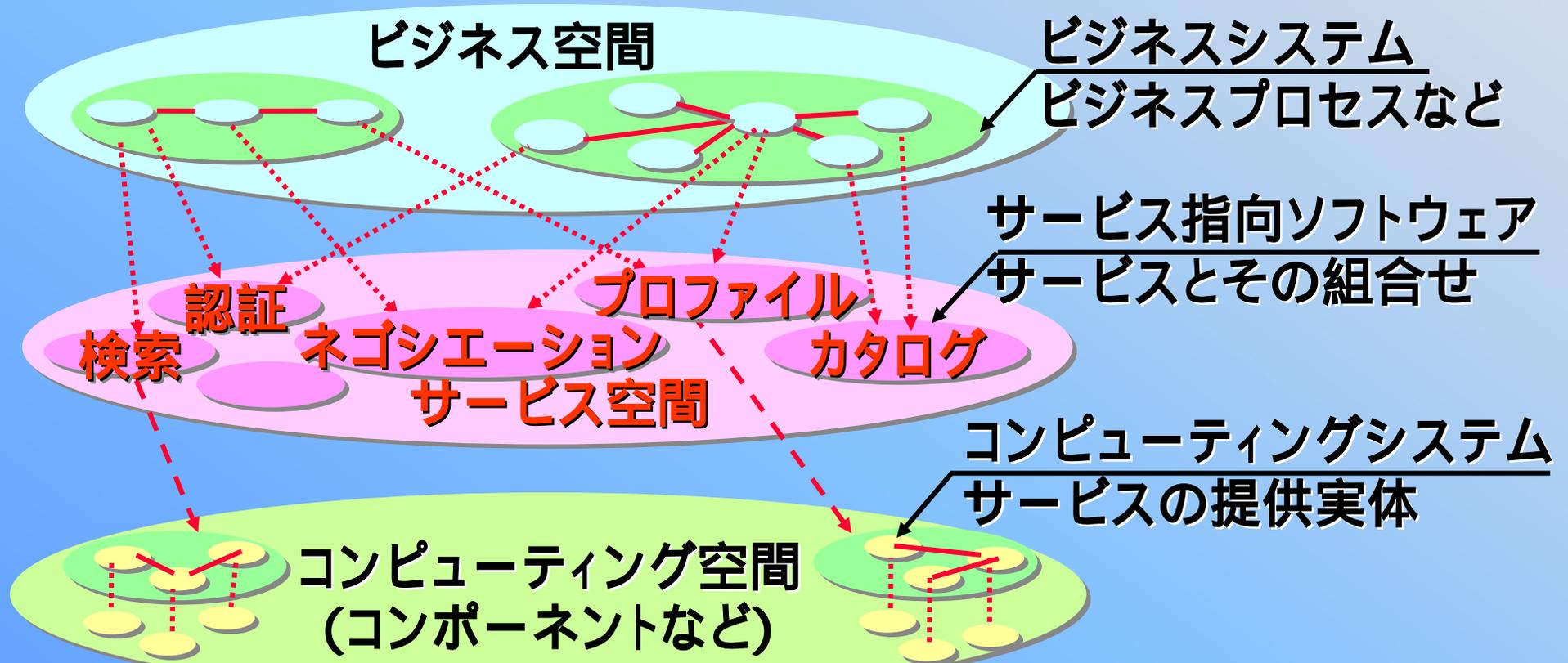
ビジネスとサービス(アーキテクチャ)とのギャップ

☞ 抽象度とアーキテクチャのギャップを埋める

☝ ビジネスゴールからSOC/SOAへの体系的展開

☞ 巨大化・複雑化しつつあるSOC/SOA技術スタック

☝ SOC/SOAの多様な技術のパッケージ化の必要性



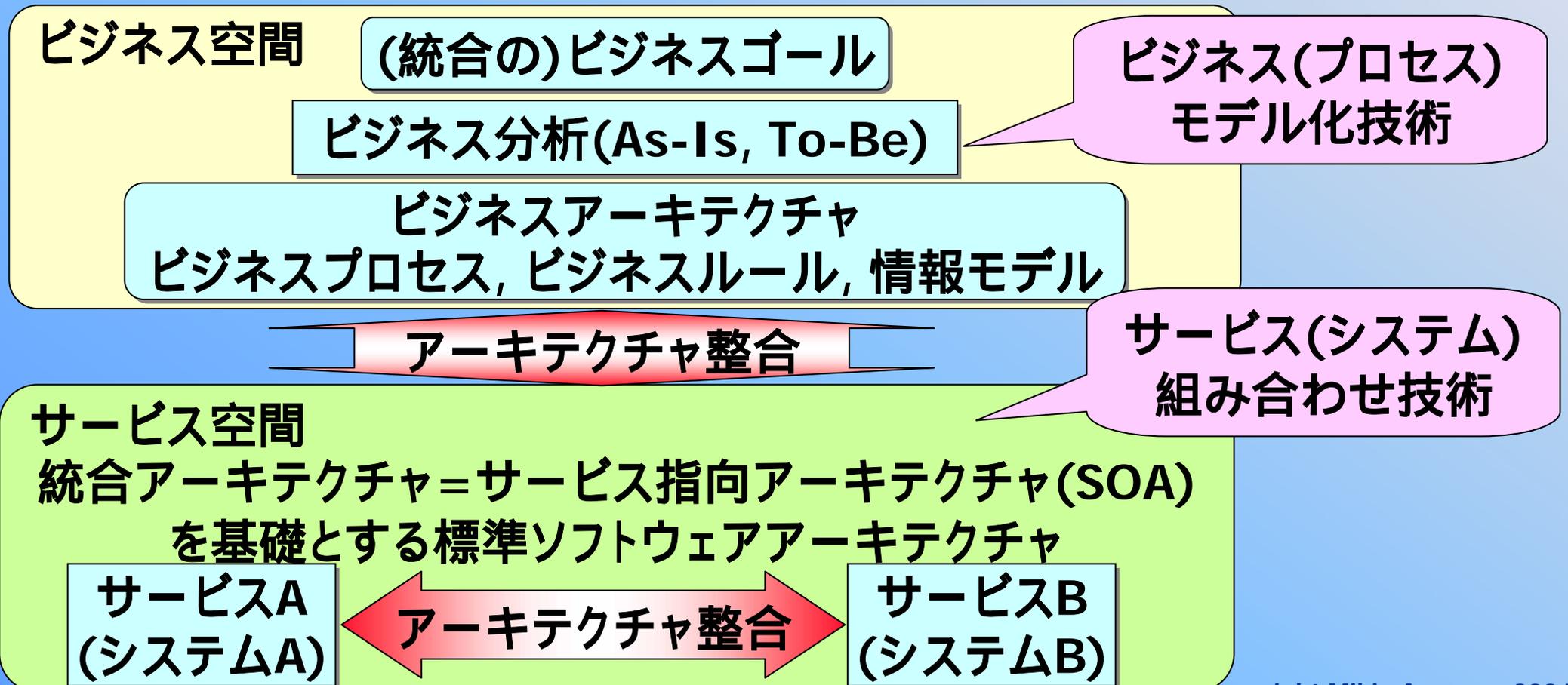
SOAはどこへ向かうのか？

サービスによるシステム統合方法論

☞ 統合方法論の必要性: 体系化が未熟

- ☞ ビジネス空間からサービス空間への整合性のある展開
- ☞ アーキテクチャ設計が鍵(プロセス, 表記法, 評価法)

☞ アーキテクチャ不整合(Architecture Mismatch)の問題



SOAはどこへ向かうのか？

サービスによるシステム統合方法論

- ☞ **ソフトウェア工学の新たな課題: 総合的アーキテクチャ設計の重要性**
 - ☞ **ビジネスゴールを実現するアーキテクチャ設計方法論**
 - ☞ **ビジネスモデル化の方法**
 - ☞ **ビジネスモデルに基づくサービス設計: 分析・設計から検証まで**
 - ☞ **動的特性の設計と検証, 動的参画のモデル**
 - ☞ **意味情報の扱い: セマンティックWebサービス**
 - ☞ **(ビジネス)オントロジー**
 - ☞ **ドメイン固有の語彙**
 - ☞ **アプリケーションとの適性**
 - ☞ **コンポーネント組み込みvsサービス組み合わせ**

SOAはどこへ向かうのか?

Webサービスによるビジネスプロセスのモデル化

👉 Webサービスを組み合わせたビジネスの実現

👉 Webサービスを組み合わせる機構の必要性

👋 **WSDL(portType):状態を持たない静的インタフェース定義**

👉 ビジネスプロセスからWebサービスへの整合性のある展開

👉 ビジネスプロセスモデリング言語(BPML)

👉 ビジネスプロセス = 実行の流れ(ワークフローモデル)が中心

👋 **Orchestration(編成), Choreography(振付け)**

👉 スコープなどの制御[BPEL]

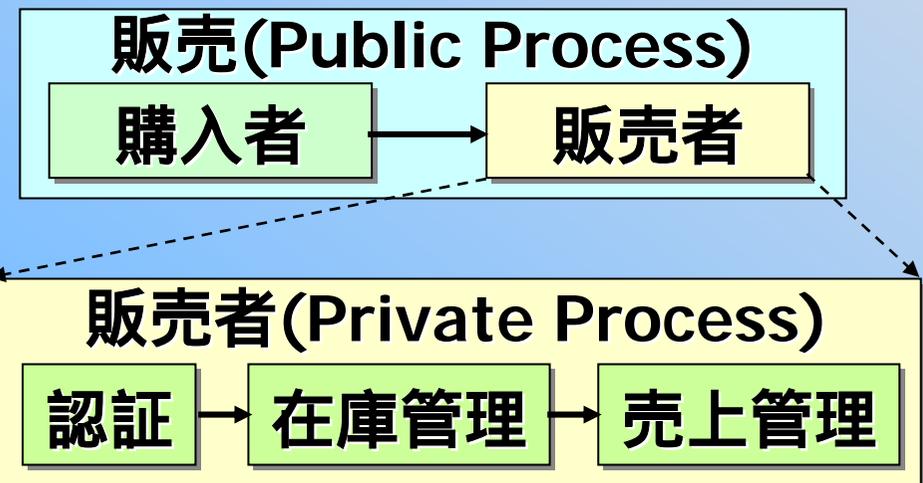
👋 **Abstract/Executable Process**

👋 **Public/Private Process**

👋 **Atomic/Complex Activity**

👉 新たな機構の必要性

👋 **動的参画**



SOAはどこへ向かうのか?

サービスを組合せたビジネスプロセス設計法

サービスを組合せたビジネスプロセス分析・設計方法

ビジネスパターンに基づく
Webサービス開発法

ビジネス分析(ビジネス空間)
パターンによるビジネスプロセス
の部品化と再利用

ソリューションダイアグラム
(ビジネス構造の表現)

ビジネス
パターン分類

統合(組合せ)
パターン分類

ビジネスプロセスの
プログラム(実行可能)記述
[BPEL]

付加価値サービスブローカ

サービスブローカ
サービスの組合せ
(設計支援+実行)

サービスディレクトリ
品質等の情報

サービス設計支援
(組合せ)

サービス仲介機構

サービス
プロバイダ
群

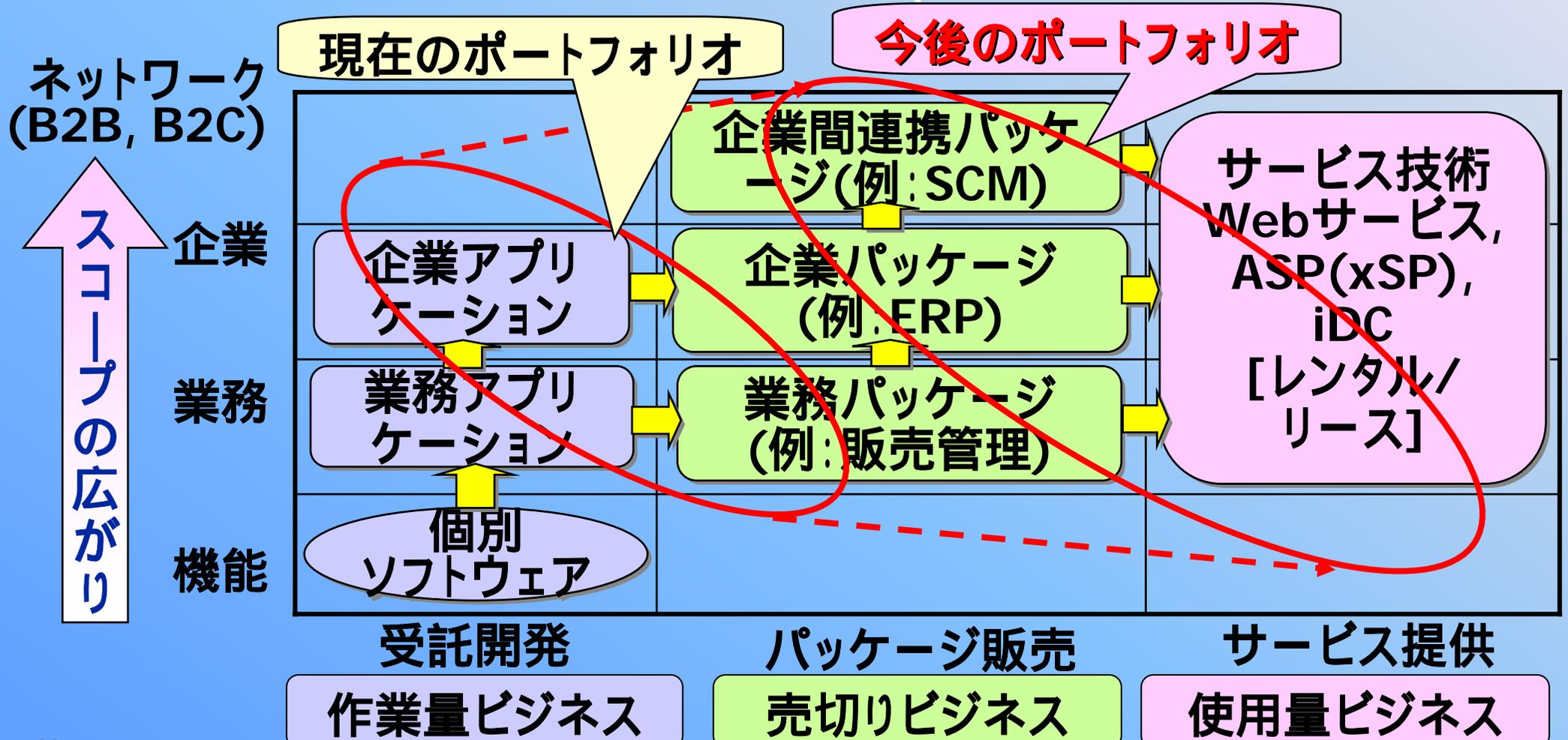
サービス
リクエスタ
群

SOAはどこへ向かうのか?

ビジネスとサービス(アーキテクチャ)とのギャップ

☞ ソフトウェアビジネスの変化

- ☞ ソフトウェアの二つの顔: プロダクトとサービス
- ☞ プロダクトからサービス化へ: SI, ソリューション



SOAはどこへ向かうのか?

Webサービス関連国内外の学会・標準化

👉 標準化: 2つのコミュニティ(?)

👉 W3C(プラットフォーム?), OASIS(アプリケーション?)

👉 研究: 2つのコミュニティ(?)

👉 Webサービス ソフトウェア工学, オブジェクト指向

👉 セマンティックWeb データモデリング&人工知能

👉 Webサービス

👉 国内: 情報処理学会ソフトウェア工学研究会 ほか, XMLコンソーシアム, 情報サービス産業協会Webサービス技術委員会[2002~3年度]
(<http://www.jisa.or.jp/committee/websrv2/>)

👉 海外:

👉 総合的会議: OOPSLA(オブジェクト指向), ICSE(ソフトウェア工学), WWW(インターネット), XML(XML)

👉 専門国際会議: ICSOC (Int'l Conf. on Service-Oriented Computing, <http://www.icsoc.org/>) [2004年11月15-19日, New York]

SOAはどこへ向かうのか?

研究開発戦略の必要性

米国からの技術輸入に依存

 日本からの技術発信は極めて稀

開発方法論の欠如

 技術の複雑度

 例: ライトウェイトなWebサービス(SOC/SOA)

新たなアプリケーションの台頭と研究開発・技術差別化の機会

 モバイルサービス

 サービスの動的発見・結合が本質的

 組み込み/ユビキタスサービス

 ネットワーク家電, 車載システムなど

まとめ

- ➡ **SOAの来た路: 技術とビジネスでI(統合)が鍵となる**
 - ☞ 分散オブジェクト環境/EAIからサービスへ
 - ☞ 独立性の高いアーキテクチャ(とインタフェース)
 - ☞ 付加価値の増大: 組織内から組織間ビジネスプロセス統合
- ➡ **SOAを活かす開発技術の必要性**
 - ☞ ビジネスからSOC/SOAへ展開できるアーキテクチャ・設計技術
- ➡ **SOC/SOAの普及**
 - ☞ SOC/SOAに適したアプリケーションと実践パターン
 - ☞ 日本的ビジネスプロセス統合のあり方=ビジネス慣行