A decorative graphic on the left side of the slide, consisting of a vertical black line intersecting a horizontal black line. The intersection is surrounded by overlapping colored squares: a purple square at the top left, a green square at the bottom left, and a blue square at the top right.

Webサービスの使いかた

Webサービスによるアプリケーション連携の真相

XMLコンソーシアム テクノロジー部会
Webサービスワークグループ

天野富夫 日本アイ・ビー・エム

テクノロジー部会WebサービスWGの活動

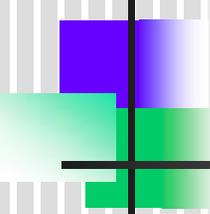
- なぜWebサービスは重要か？
- どんなメリットがあるのか？
- どう作り、使えばよいのか？

事例の収集を行っているが

- メリットの定量的評価は困難
- ノウハウは外部には出にくい
- オースライズされた情報の不足

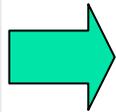
そこで...

- ◆ 事例から納得できるロジック(仮説)を集める
- ◆ 仮説を検証する
(他の仮説やITのパターンと整合しているか)
- ◆ 仮説を統合し一般化する

A decorative graphic consisting of a purple square and a green square overlapping, with a black crosshair-like structure.

Webサービスの使いかた

■ 内容



1. Webサービスのもたらすメリット
2. Webサービスの再定義
3. Webサービスによるアプリ連携
4. まとめ

Webサービスのもたらす**はずの**メリット

どのように利益を生み出すかで3つのパターンに分類

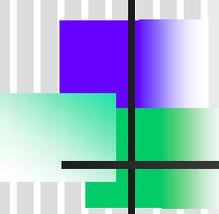
(WebサービスWGの活動 2001～2002、2002年コンソーシアムWeek)

- アプリケーション連携/プロセス統合
 - 社内/パートナー(顧客)企業との密接な連携
 - 新たな付加価値(品質・機能)を顧客に提供
 - コスト削減
- 機能提供
 - 機能を提供し対価を得る
 - Webサービス使用料
 - 販売促進/新規ビジネスパートナーの獲得
- 仲介・支援
 - 取引/連携を行うための仲介
 - 与信/拡張検索
 - プロビジョニング(2003年第4回コンソーシアムDay)

Webサービスによるアプリケーション連携

- ビジネスはよりダイナミックに。“Time-to-market”が至上命題
- 新しいアイデア(=イノベーション)には新しいビジネスプロセスが必要
- 既存アプリケーション機能をソフトウェア部品化して連携させビジネスプロセスを実現

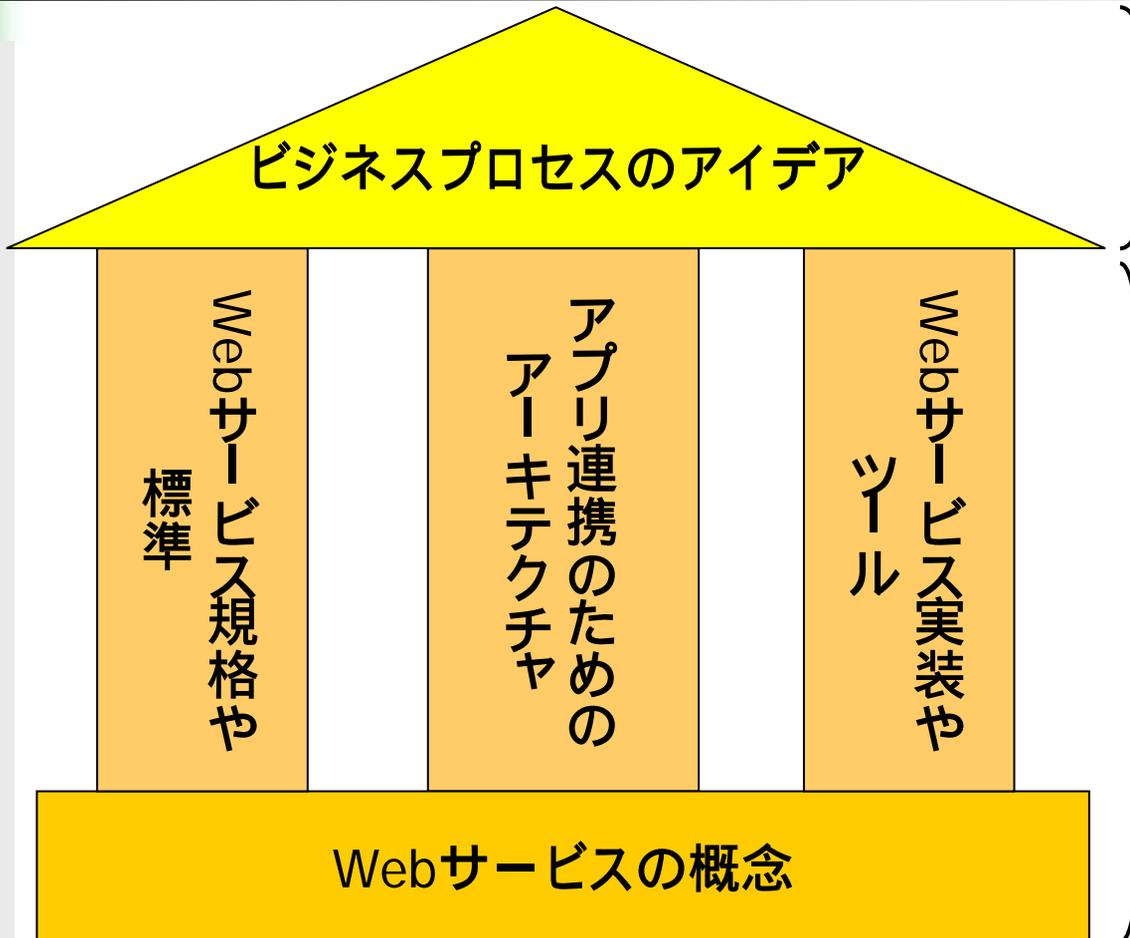
Webサービス事例	内容/メリット
ECサイトからメインフレーム側在庫管理システムへの照会を可能に。	<ul style="list-style-type: none"> ■顧客への注文確認待ち時間の短縮(e.g. 1.5日 20分) ■在庫切れに対する迅速な対応
年金計画のデータ入力インタフェースをWebサービスとして公開。顧客企業の給与システムと連携。	<ul style="list-style-type: none"> ■給与データ入力(50人の専任オペレータ)のコスト削減
研究用に作成したプログラム群をWebサービスで連携	<ul style="list-style-type: none"> ■シミュレーションの自動化、生産性の向上
個別のサイトで提供されていた教育コースの情報を集約	<ul style="list-style-type: none"> ■One Stopサービスの提供(アグリゲーションサービス) ■新規付加価値の実現



Webサービスのビジネス形態

- 主流はアプリケーション連携/プロセス統合
- ビジネス上のアイデア(価値)の迅速な実現
 - 既存ソフトウェアを部品として有効活用
 - 他者が提供する機能の活用
 - 機能提供のビジネス
 - Webサービスによるプロセス統合の支援
 - 仲介、プロビジョニング

Webサービスのメリット = 実現したビジネスプロセスのメリット

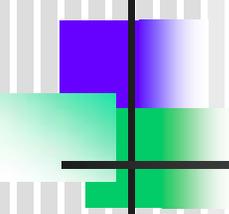


ビジネスのアイデアから
メリットが生み出される

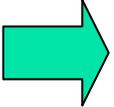
本日のテーマ

どんなふうにWeb
サービスを使えばよ
いのか？

- Webサービスによる迅速な実現
- Webサービスを利用して初めて実現できるプロセス

A decorative graphic in the top-left corner consisting of a purple square, a green square, and a black crosshair.

■ 内容

1. Webサービスのもたらすメリット
-  2. Webサービスの再定義
3. Webサービスによるアプリ連携
4. まとめ

WebサービスとはSOAPで接続すること？

- SOAPでつなぐと何がうれしいのか？
既に確立している接続をSOAPで作直しても意味がない
従来技術と比べてパフォーマンスはどうなの？

- ◆ 「Webサービス = SOAP」という図式の検証
- ◆ 既存の接続形態でWebサービスができること
- ◆ あえてSOAPを使うことの意義

そもそもWebサービスとは？

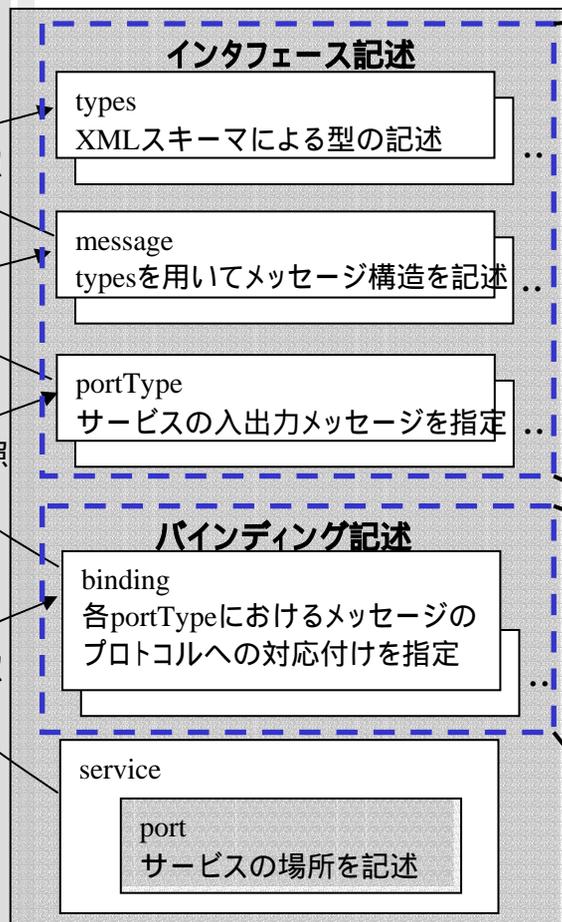
- Webサービスとは、
 - A) URIによって識別され、
 - B) その**インターフェースおよびバインディング情報**は、XMLによって定義・記述され、
 - C) それらの情報は他システムから検索が可能であり、
 - D) インターネット・プロトコル上の
 - E) XML書式のメッセージを使用して、他のWebサービスと通信する、
- ソフトウェアシステムである
- 定義自体は特定の規格と結びついたものではない

出典： W3C WebサービスアーキテクチャWG

<http://www.w3.org/TR/2003/WD-ws-arch-20030514/>

XMLによるインタフェース/バインディング記述

WSDL文書の構造



WSDLの記述例

```

<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="TradePriceRequest">
      <complexType>... </complexType>
    </element>
  </schema>
</types>

<message name="GetLastTradePriceInput">
  <part name="body" element="xsd1:TradePriceRequest"/>
</message>

<portType name="StockQuotePortType">
  <operation name="GetLastTradePrice">
    <input message="tns:GetLastTradePriceInput"/>
    <output message="tns:GetLastTradePriceOutput"/>
  </operation>
</portType>

```

```

<binding name="StockQuoteSoapBinding" type="tns:StockQuotePortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetLastTradePrice">
    <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>

```

WSDL仕様によれば...

- WSDL中のインタフェース記述には抽象レベルのメッセージフォーマットが定義されている
- 具体的なフォーマットはバインディング記述によって決定される(ただし抽象から具体へのマッピングの手続きの詳細はWSDLには記述されていない、パラメータのみ)
- ある種のバインディング記述(SOAP/HTTPバインディング)では具体フォーマットとして抽象フォーマットがほぼそのまま使われる

2.2 Message

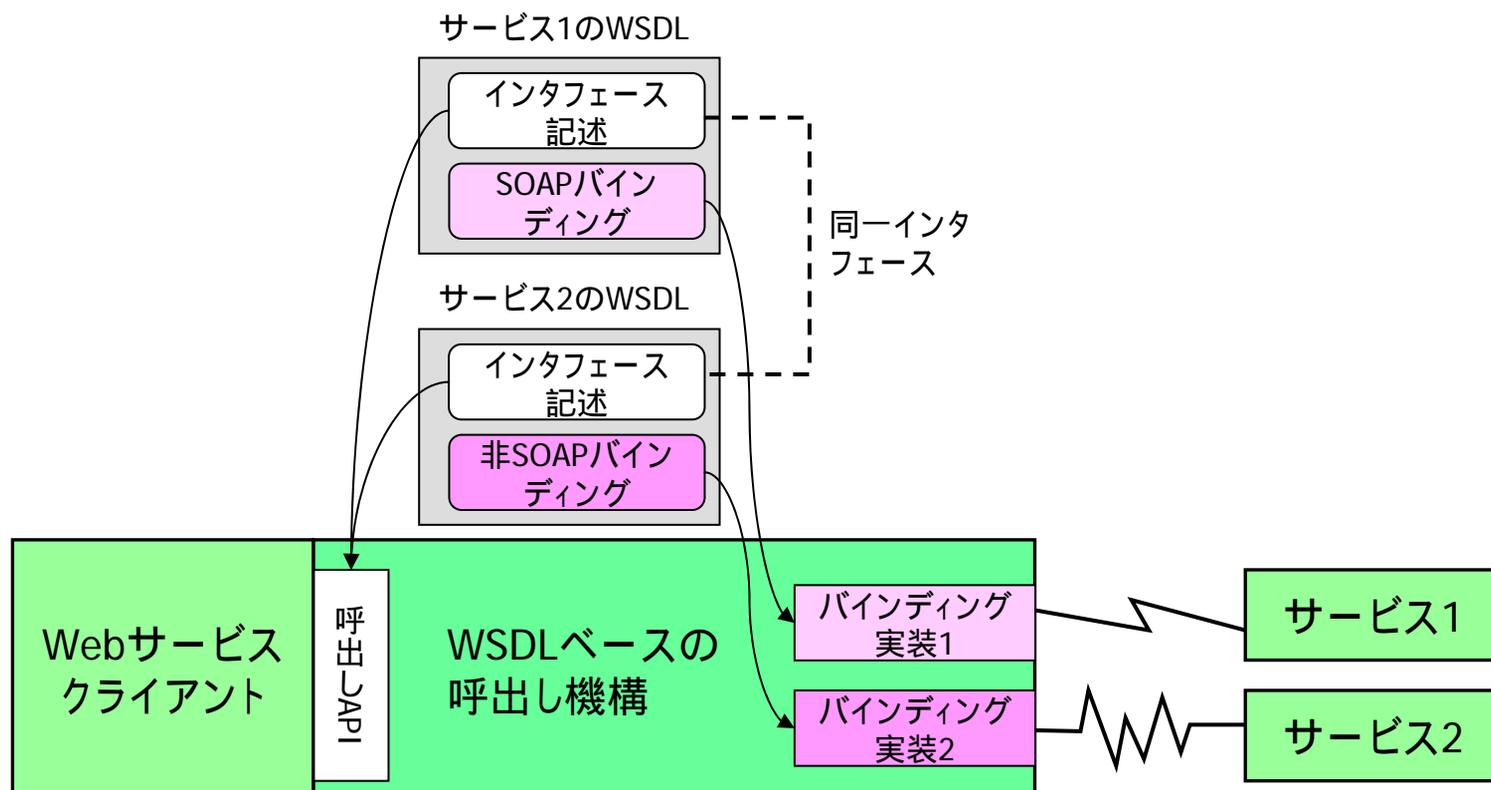
2.2.1 The Message Component

A message component describes the **abstract format** of a particular message that a Web service sends or receives. The format of a message is typically described in terms of XML *element information items* and *attribute information items*. A **message binding** (see 2.7 Binding [p.21]) describes **how the abstract content is mapped into a concrete format**. However, in some cases, the abstract definition may match the concrete representation very closely or exactly for one or more bindings. Such bindings will supply little or no mapping information. However, another binding of the same message definition may require extensive mapping information. For this reason, it is not until the binding is inspected that one can determine "how abstract" a message really is.

WSDL 1.2 3月3日版ワーキングドラフト(<http://www.w3.org/TR/2003/WD-wsdl12-20030303>)より

WebサービスはWSDLで呼び出される

- オープンソースやITベンダーによるWSDLツールの提供
- バインディングの違いはクライアント側からは隠蔽される



本質はXMLによるインタフェース記述

SOAPを使って
いなくても

インタフェースが
WSDLで記述さ
れていれば

Webサービスでなくても
Webサービスのように呼び出せれば
Webサービスである。

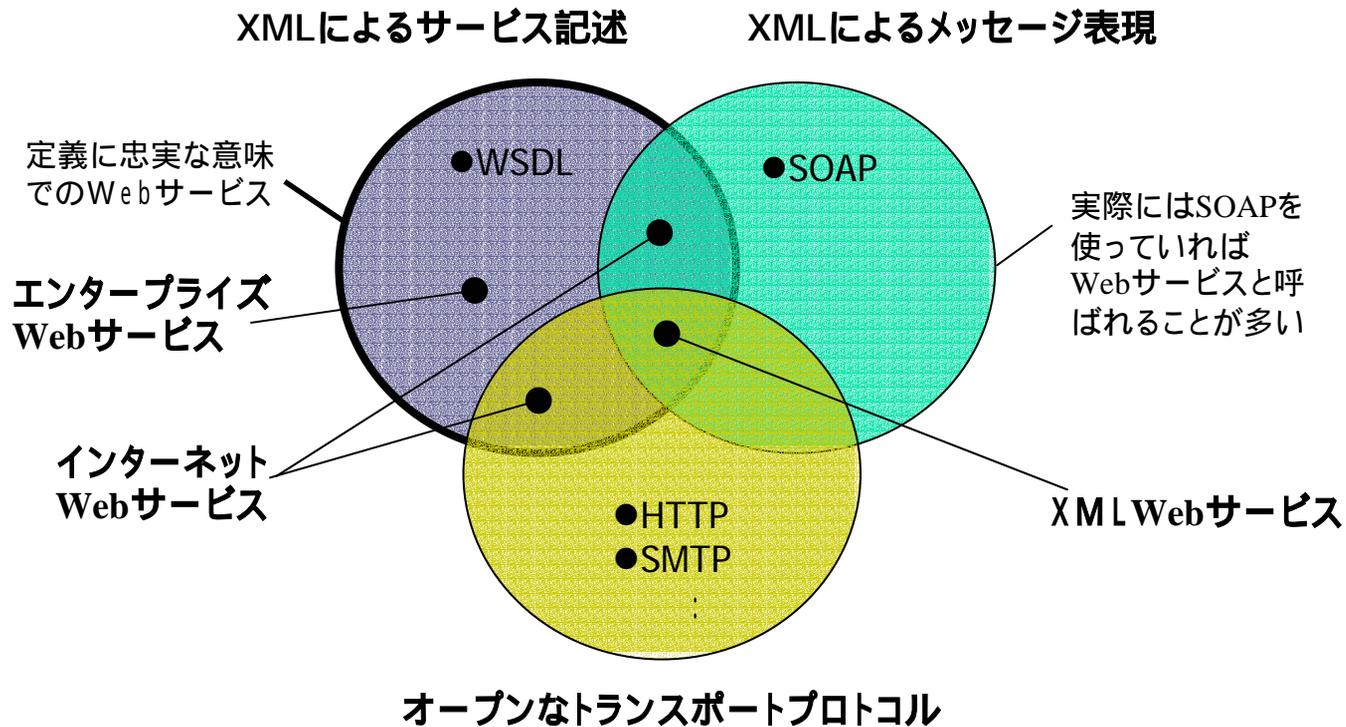
- 同じメカニズムを使って呼び出すことができる

Webサービスの分類

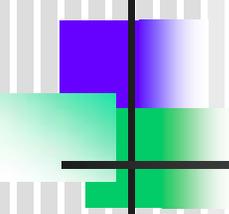
- **エンタープライズWebサービス**: WSDL記述を備えていながら、プロプライエタリーなアプリケーション・メッセージングまたはトランスポート・プロトコルを使用する可能性のあるWebサービス
- **インターネットWebサービス**: オープンなアプリケーション・メッセージングまたはトランスポート・プロトコルのみを使用しなければならないエンタープライズWebサービス
- **XMLWebサービス**: HTTPやSMTPなどのトランスポート・プロトコルを介して、W3Cが採用したXMLベースのメッセージング・プロトコルを使用するインターネットWebサービス

参考: http://www-6.ibm.com/jp/developerworks/webservices/021206/j_ws-best1.html

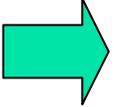
Webサービスの分類



参考: http://www-6.ibm.com/jp/developerworks/webservices/021206/j_ws-best1.html

A decorative graphic in the top-left corner consisting of a purple square and a green square overlapping, with a black crosshair.

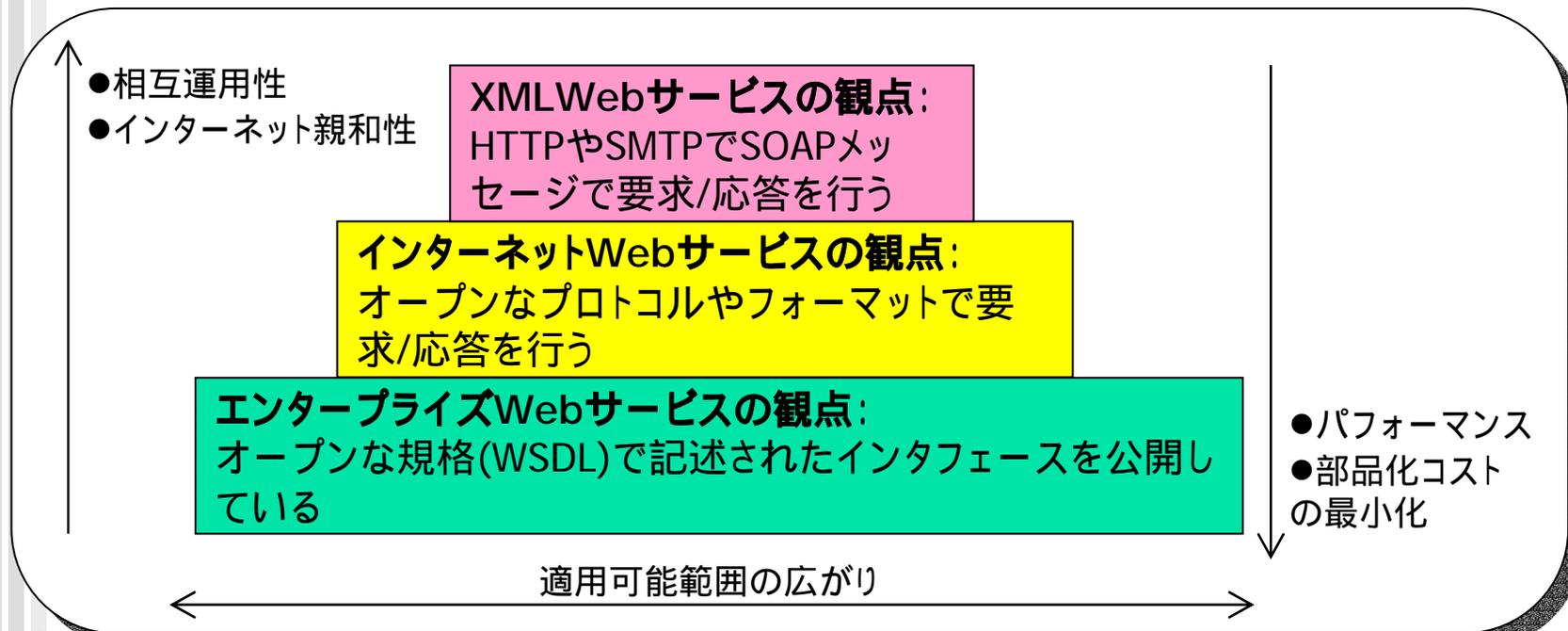
■ 内容

1. Webサービスのもたらすメリット
2. Webサービスの再定義
-  3. Webサービスによるアプリ連携
4. まとめ

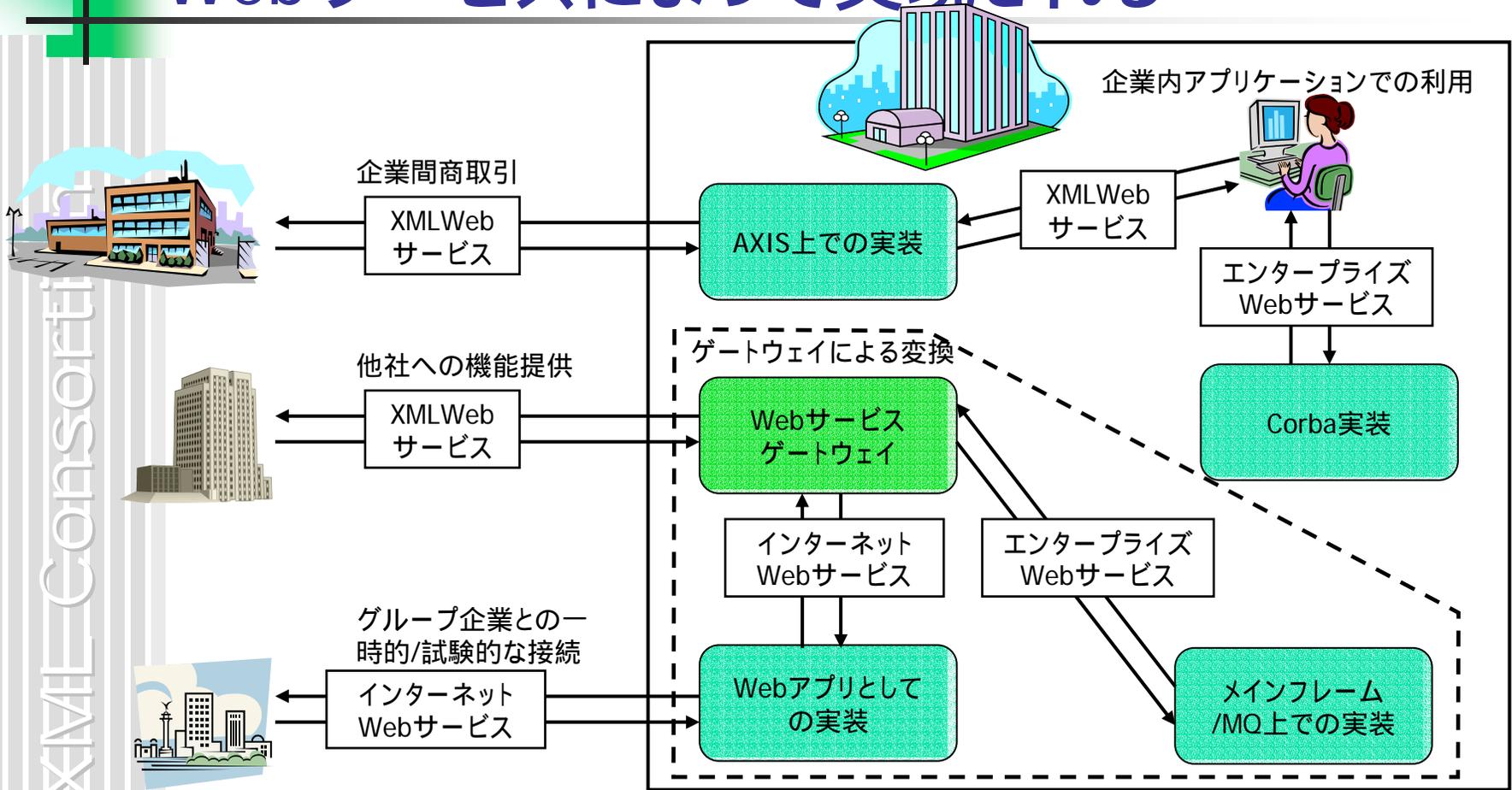
Webサービスによるアプリケーション連携

- ビジネスはよりダイナミックに。“Time-to-market”が至上命題
- 新しいアイデア(=イノベーション)には新しいビジネスプロセスが必要
- 既存アプリケーション機能をソフトウェア部品化して連携させビジネスプロセスを実現

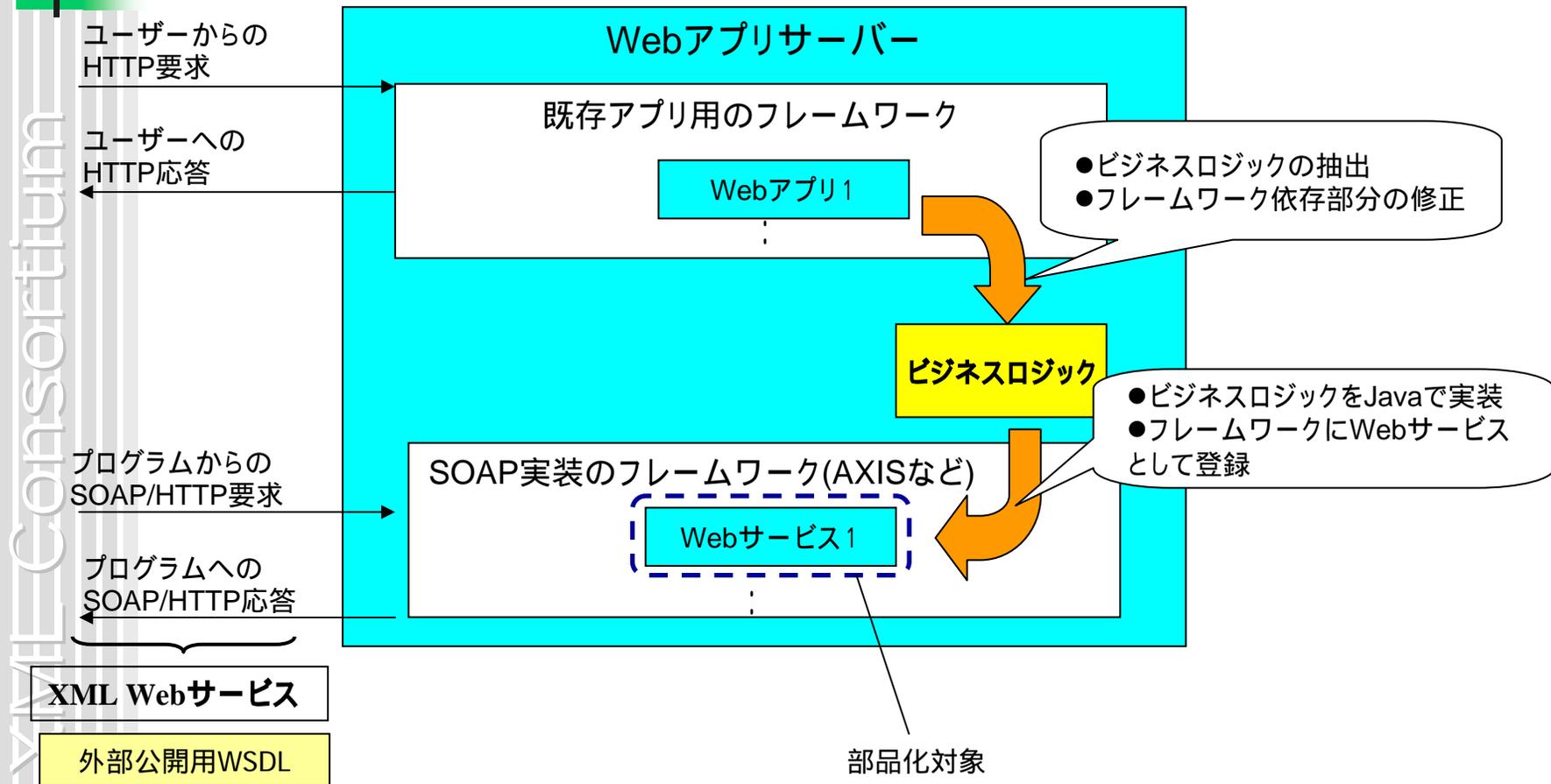
既存機能のソフトウェア部品化とは



End-to-Endのアプリ連携は複数のタイプのWebサービスによって実現される



XML Webサービスとしてのソフトウェア部品化例



XMLWebサービスによる部品化

メッセージレベルの相互運用性・互換性

- 異なる分散フレームワーク間を接続する”のり”
 - EAIミドルウェアのためのミドルウェア

インターネット親和性

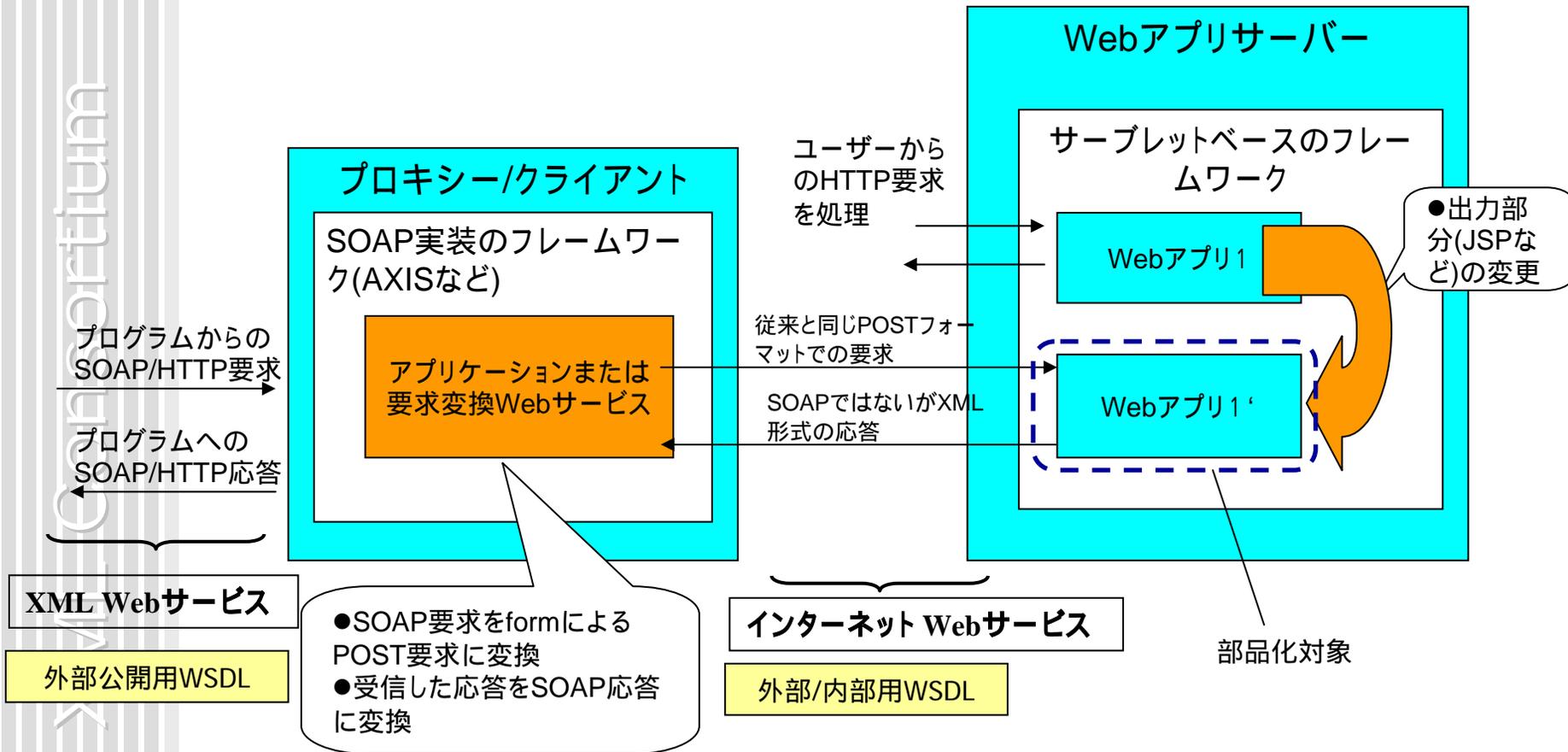
- 外部公開用サービス

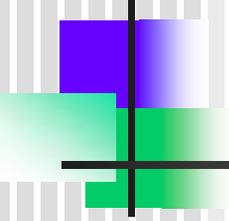
他の規格を適用できない局面で使うのが本道

- ✘ パフォーマンスはプロプラな規格に劣る
- ✘ 提供側にWebアプリケーションサーバーが必要
- ✘ ロジックを抽出しSOAP実装に配備するためにワークロードが発生

(ロジックがEJBなどで既に部品化されている場合はツールによる対応が可能)

インターネットWebサービスとしてのソフトウェア部品化例





インターネットWebサービスによる部品化

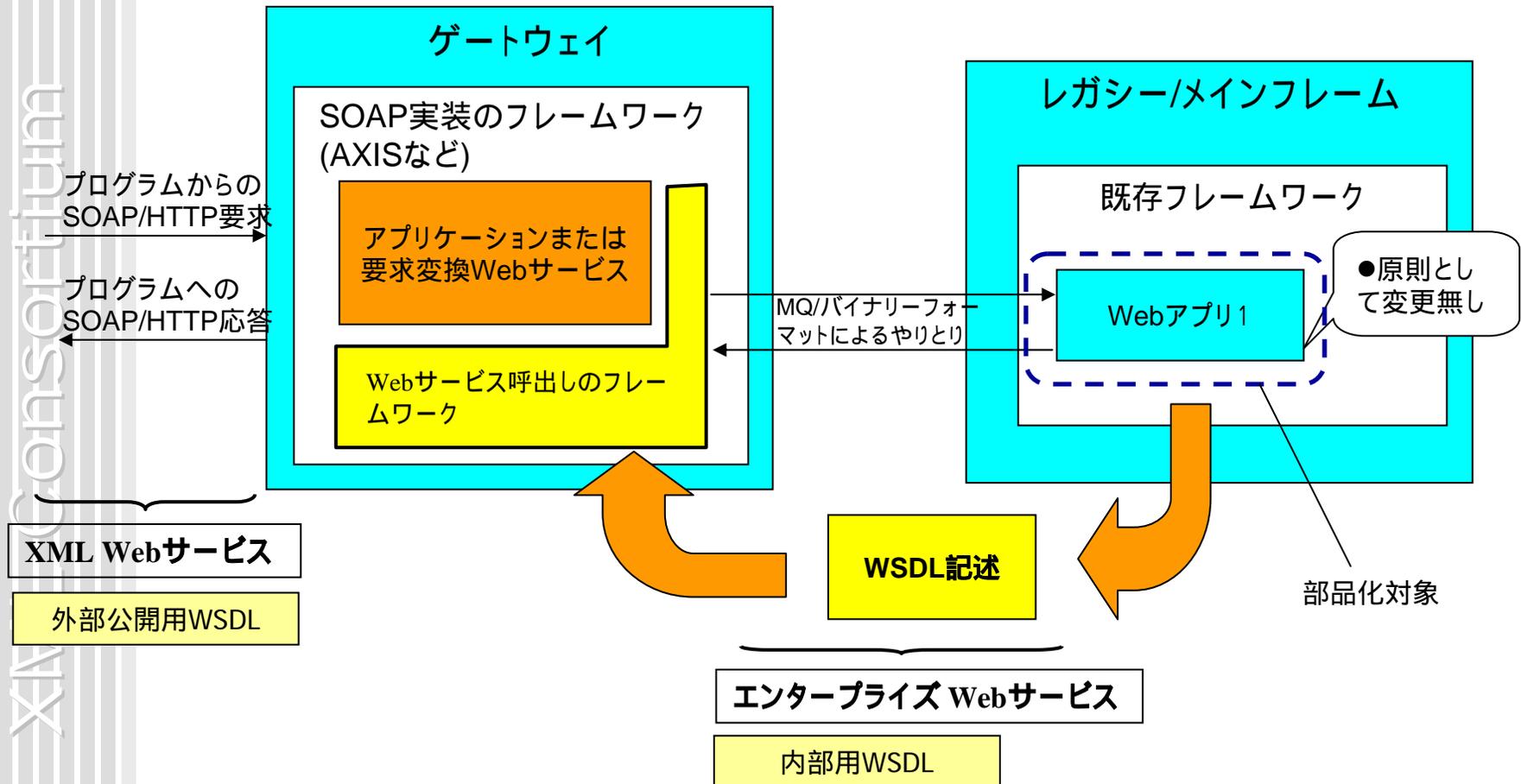
Webアプリケーション実装との連続性

- 既存Webアプリケーション変更の最小化

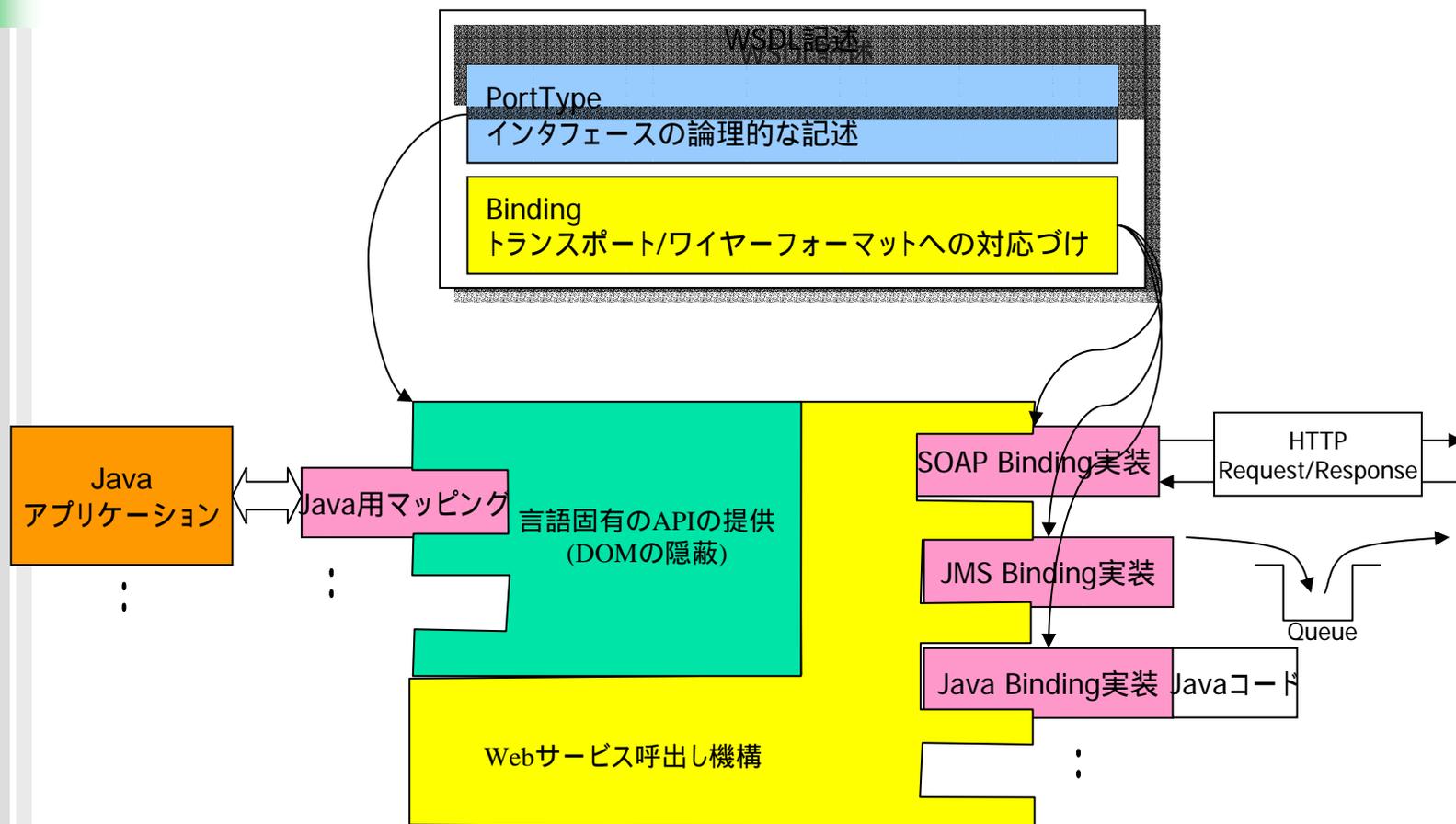
プロキシ/ゲートウェイによりXMLWebサービスへ変換可能

- ✗ Webアプリケーションとして実現されている機能が対象
- ✗ アドホックな実装(Screen scraping、セッション情報のセットアップ等)が必要になるときがある
- ✗ パフォーマンスはプロプラな規格に劣る

エンタープライズWebサービスとしてのソフトウェア部品化例



Webサービス呼出しフレームワーク



例えば、Java用のWSIF (<http://ws.apache.org/wsif/>)

エンタープライズWebサービスとしての部品化

既存実装(レガシー)への変更は最小
提供側でWebアプリケーションサーバー不要
パフォーマンスはほとんど低下しない

プロキシ/ゲートウェイによりXMLWebサービスへ変換可能

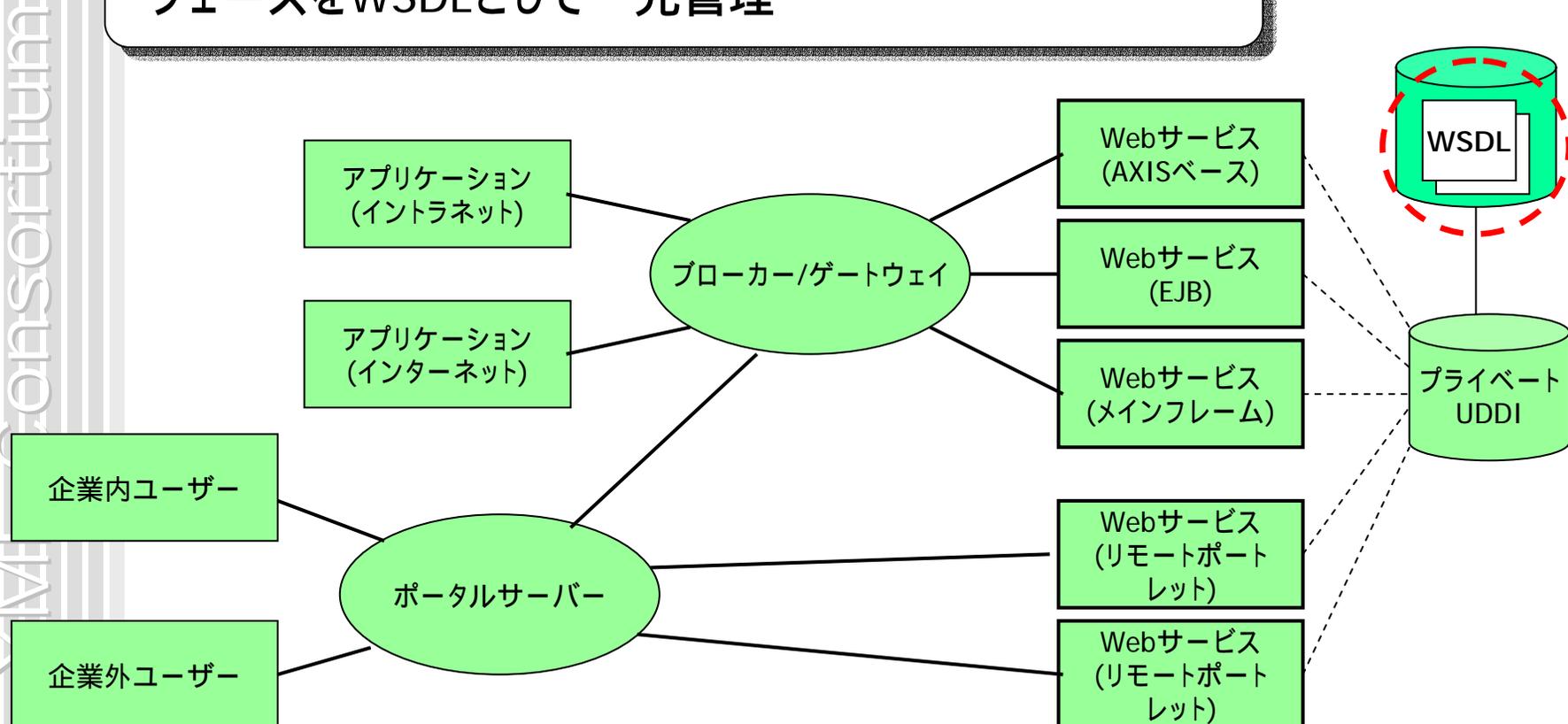
- ✘ メッセージ/トランスポートレベルの互換性・相互運用性は無い
 - ✘ 呼出し側のインタフェースレベルで互換性を確保
 - ✘ WSIFのような呼出し機構が必要
 - ✘ プラグインの実装への依存性

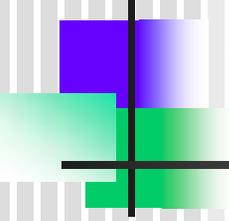
WSDL定義のメリット

- WSDLからコードの自動生成
 - サーバー/クライアント双方のコード生成
 - ケアレスミスの回避
 - SOAP実装依存の呼出しコードの隠蔽
- 実装からWSDLの自動生成
 - 既存実装のWebサービス化
 - Javaコードから
 - JCAのリソースアダプターから
- 全てのサービスのインターフェースの一元記述
 - メッセージの論理フォーマットの分離
 - 物理(ワイヤー)フォーマットは別に定めてもよい
 - トランスポートの分離
 - 実装変更時のスムーズな移行
 - XMLワークフロー記述から呼び出せる
 - BPEL4WSの記述はWSDLによるPortType等の定義を参照

企業内IT機能の一元管理

プラットフォームやプロトコルが異なるIT機能のインタフェースをWSDLとして一元管理

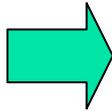


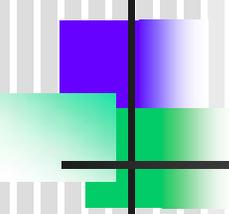


Webサービスの使いかた

■ 内容

1. Webサービスのもたらすメリット
2. Webサービスの再定義
3. Webサービスによるアプリ連携
4. まとめ



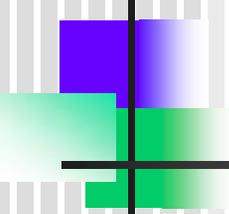


まとめ -噂の真相-

- Webサービスの基本はXMLによるインタフェースの記述である
 - SOAP/HTTPで通信することではない
 - (旅行予約を代行してくる)知的エージェントでもない
- アプリ連携/プロセス統合とはインタフェースの記述方式の統一である
 - ワイヤレベルのフォーマット/プロトコルの統一ではない

まとめ -次に考えるべきこと-

- 素早い統合のための方法論
 - WebサービスツールやIDEの活用が必須
 - 手作りのWebサービス実装はあまり意味がない
 - WSDLレベルの相互運用性をどう確保していくか
- Webサービスを利用するためのアーキテクチャ
 - 自動化できる部分/ツールを使える部分の見極め
 - 部品化の戦略の選択
 - 既存の接続のSOAPによる再実装は無意味
 - 再利用性とパフォーマンスのトレードオフ
 - 相互運用性とパフォーマンスのトレードオフ
 - パフォーマンスと再利用性のトレードオフ
 - 他の技術動向(例えばJ2EE)との関連

A decorative graphic consisting of a purple square, a green square, and a black crosshair.

Webサービスに興味をお持ちですか？

- Webサービスは確実なITの潮流です
- 原理は単純ですが、使いこなすためには知るべきことがいろいろあります
- WebサービスWGでいっしょに探っていきませんか？