

電子商取引 (EC)

~ 次世代 ECのプロットフォームとして
期待されるWebサービス基盤 ~

2002年5月27日

日本電気株式会社

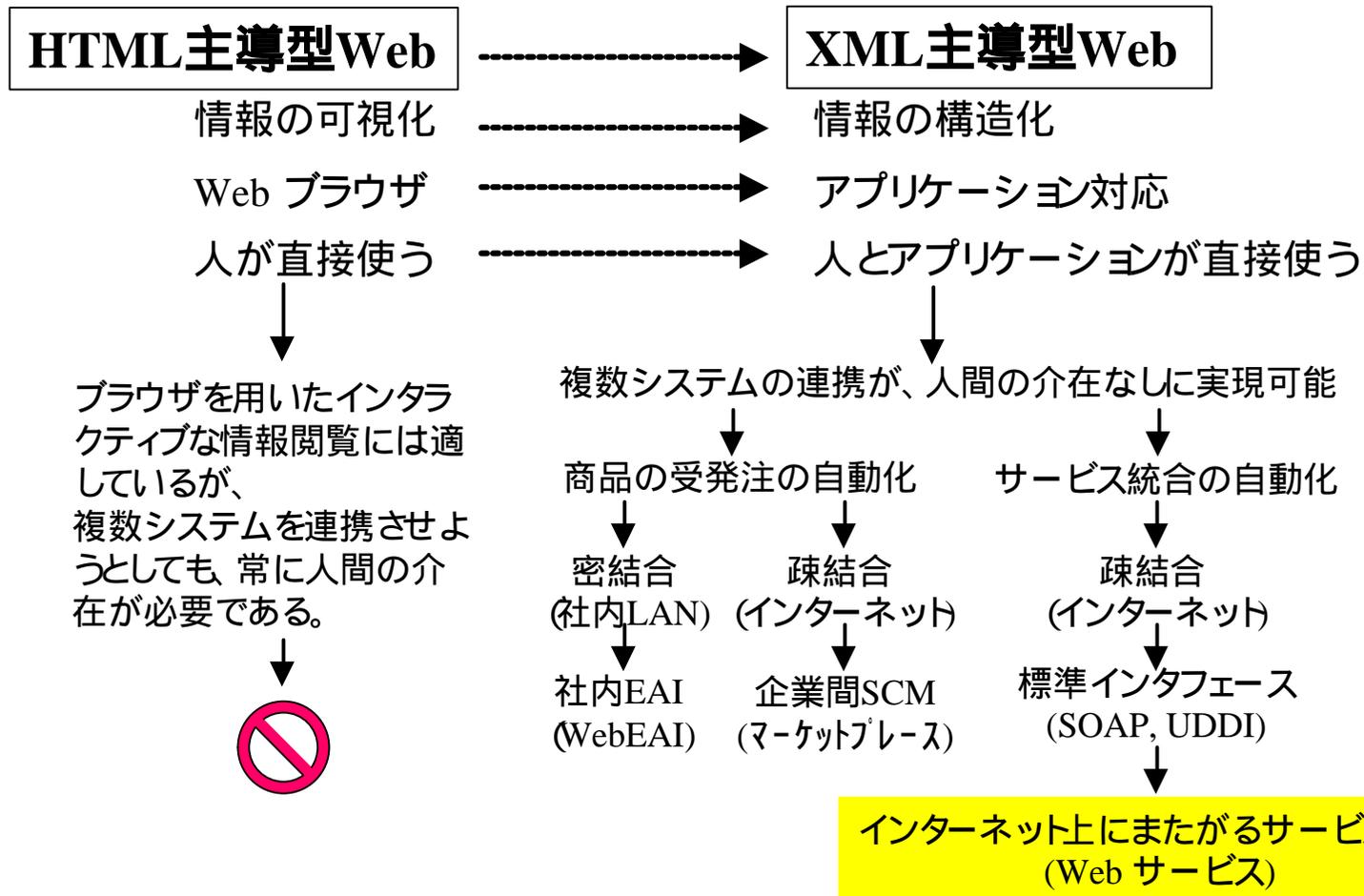
山之内 徹

講義内容

1. Webサービスとは何か
2. Webサービスを支える技術
3. Webサービスのビジネス
4. Webサービスの業界動向
5. NECの取り組み
6. Webサービス発展のステップ
7. Webサービスの課題
8. まとめ

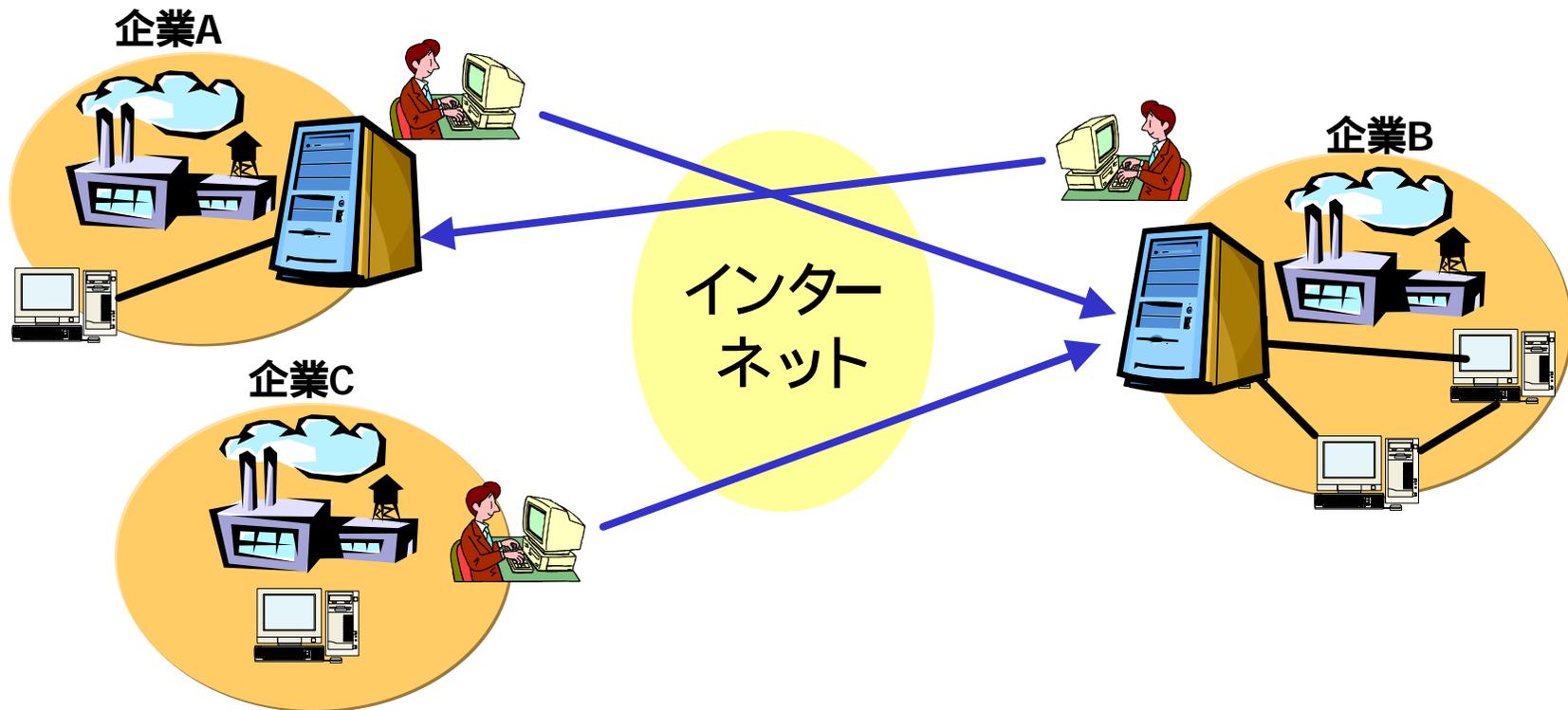
1. Webサービスとは何か

WebブラウジングからWebサービスへ



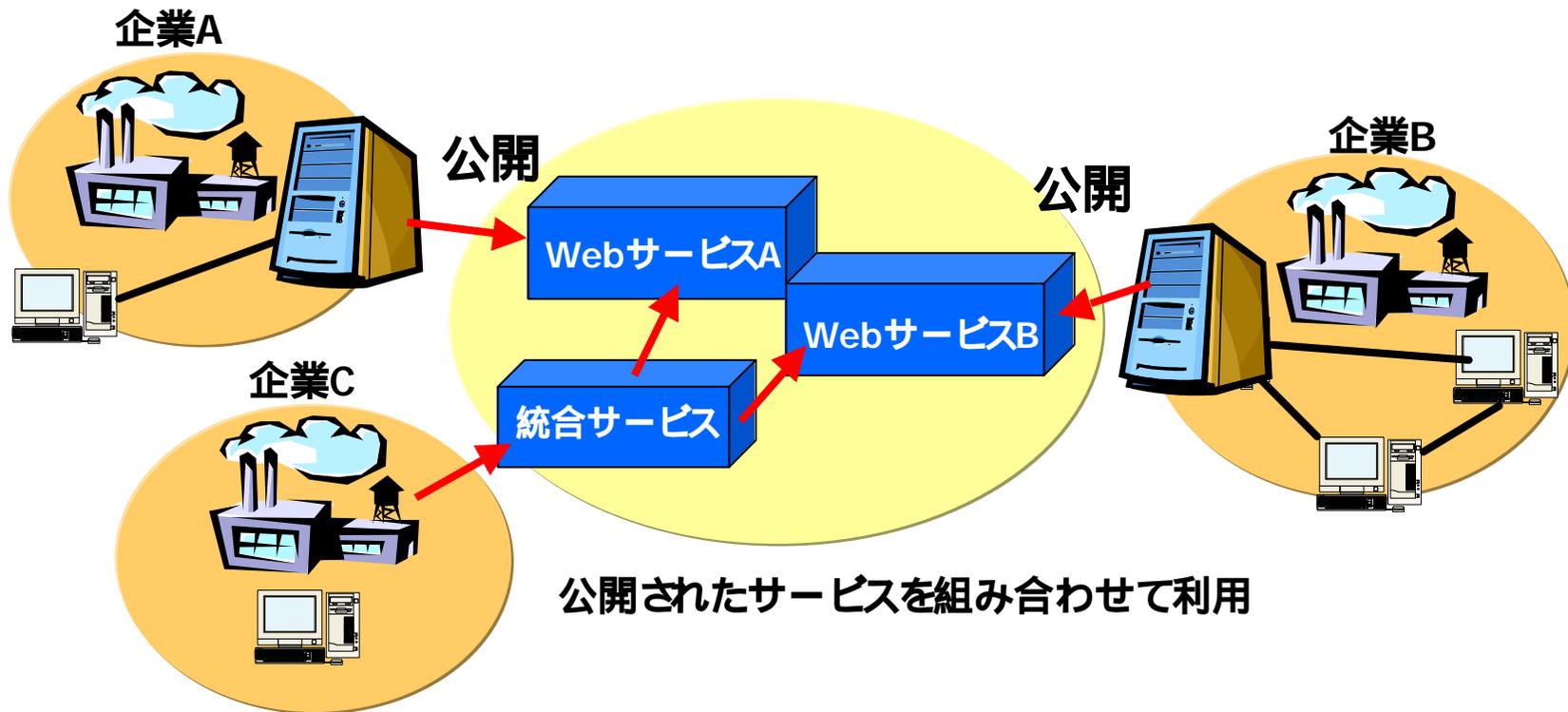
1.1 Webサービスとは

インターネット上の標準プロトコルを使用したWebサーバ上のコンポーネントであり、これらを組み合わせて新たなサービスを構築



1.1 Webサービスとは

インターネット上の標準プロトコルを使用したWebサーバ上のコンポーネントであり、これらを組み合わせて新たなサービスを構築



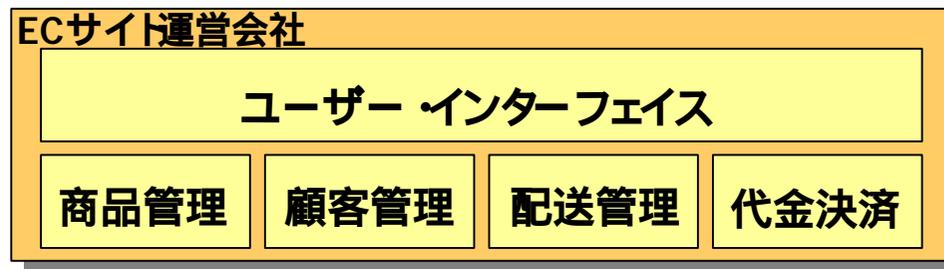
新たなサービスを動的に構成

Just-in-time Integration

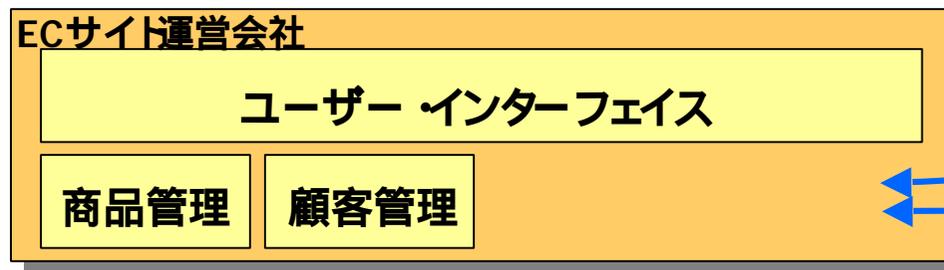
Webサービスによるシステム (1)

ASPがインターネットに公開したサービスをシステム構築に利用

従来のシステム



Webサービスを利用



ASPが公開する
Webサービスを利用

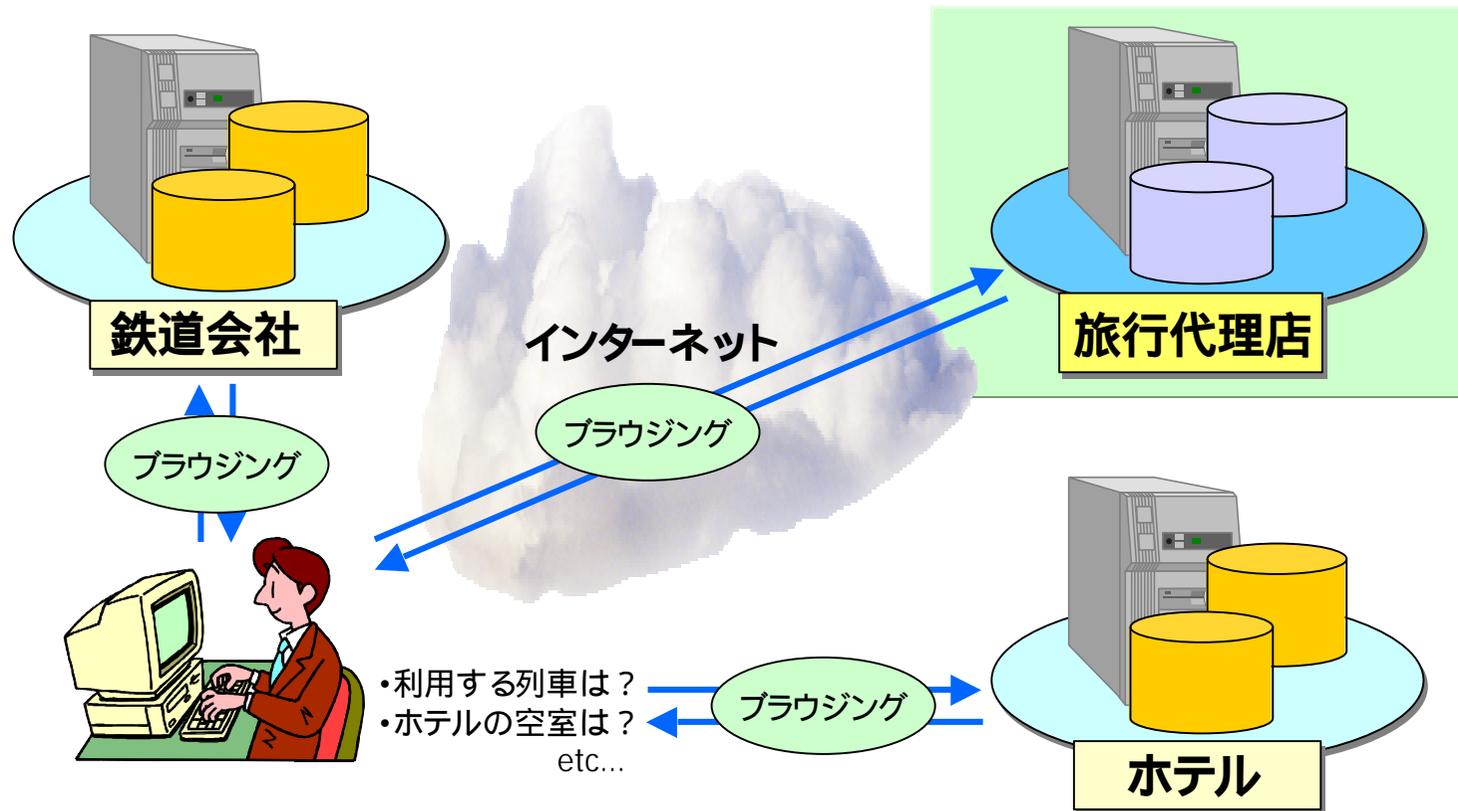
インターネット

物流ASP
配送管理

決済ASP
代金決済

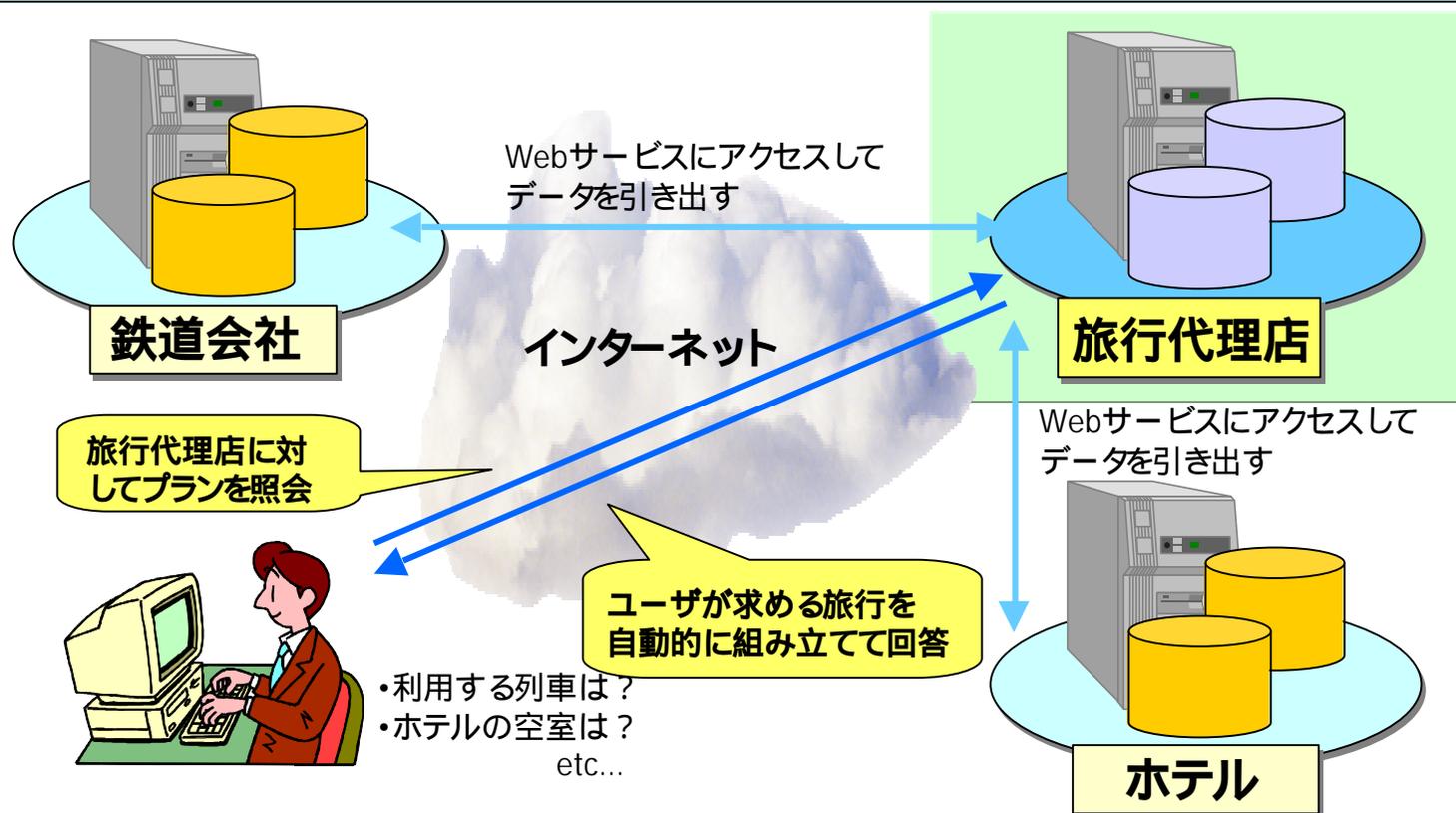
Webサービスによるシステム (2)

Webサービス間で連携がとられ、統合サービスとして
利用ユーザに対する最適解が得られる



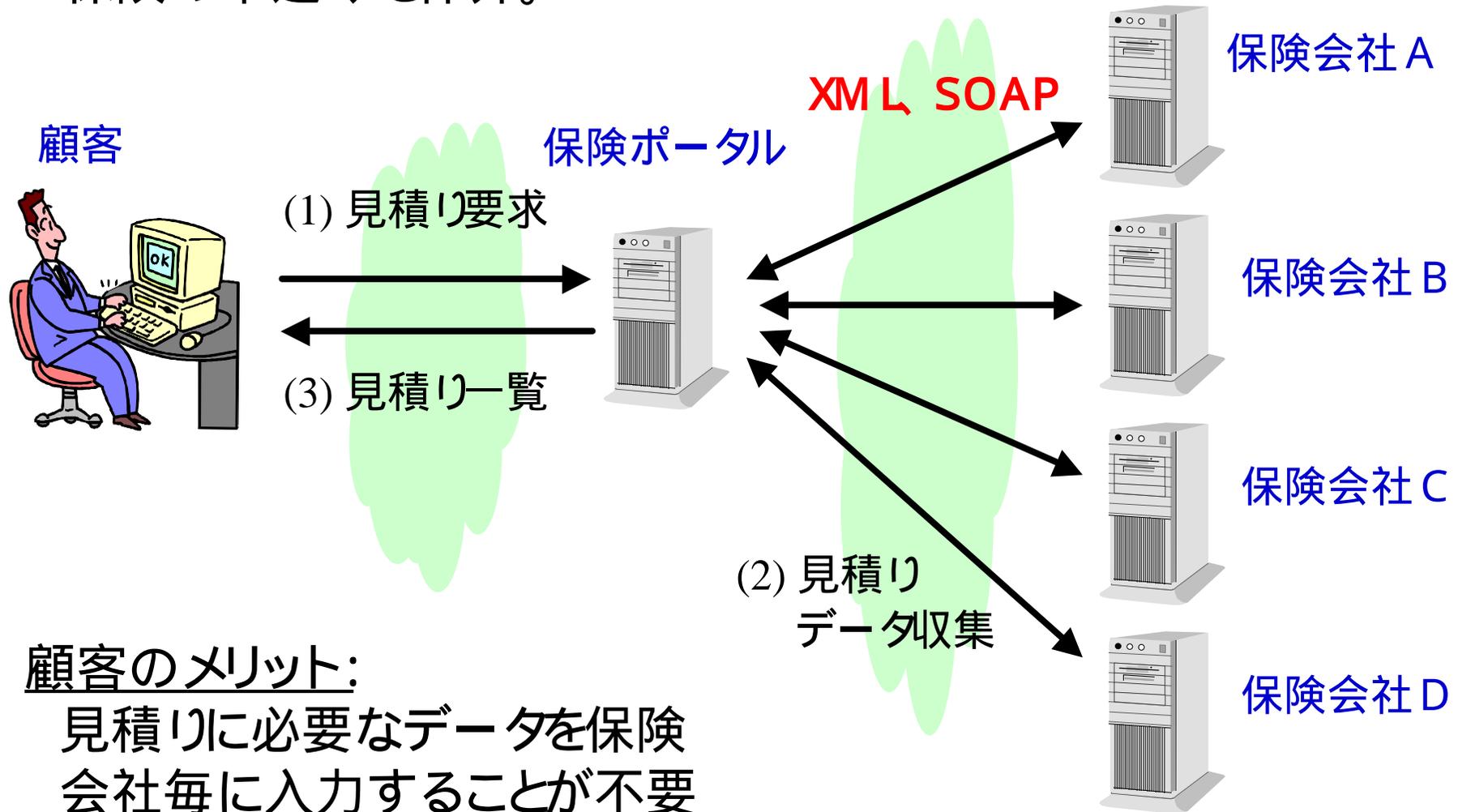
Webサービスによるシステム (2)

Webサービス間で連携がとられ、統合サービスとして
利用ユーザに対する最適解が得られる



1.2 Webサービスの事例 (1): 保険ポータル

多数の保険会社からの見積もりを取りまとめて顧客に提示し、
保険の申込みも仲介。

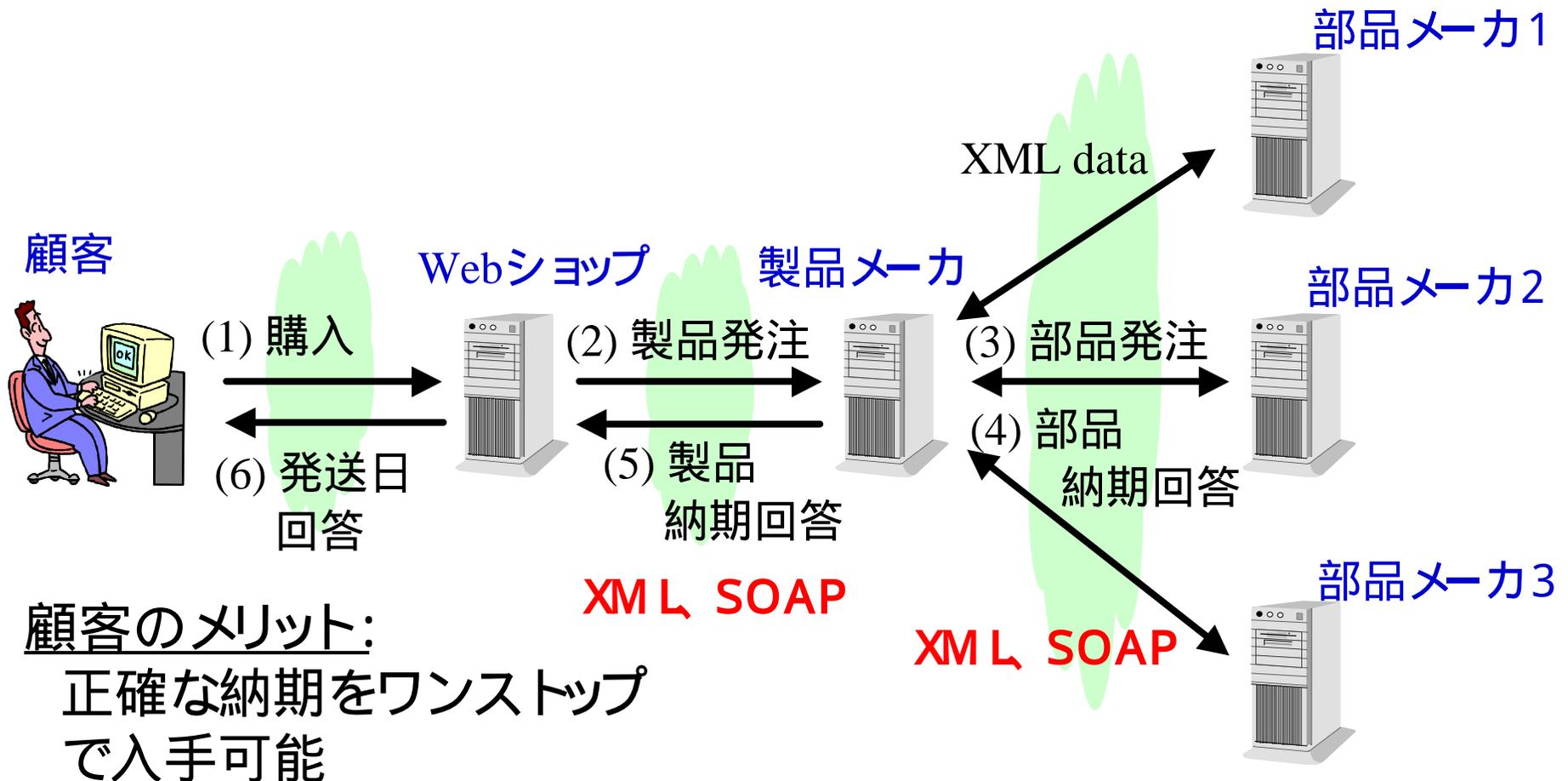


顧客のメリット:

見積りに必要なデータを保険会社毎に入力することが不要

Webサービスの事例 (2): SCM

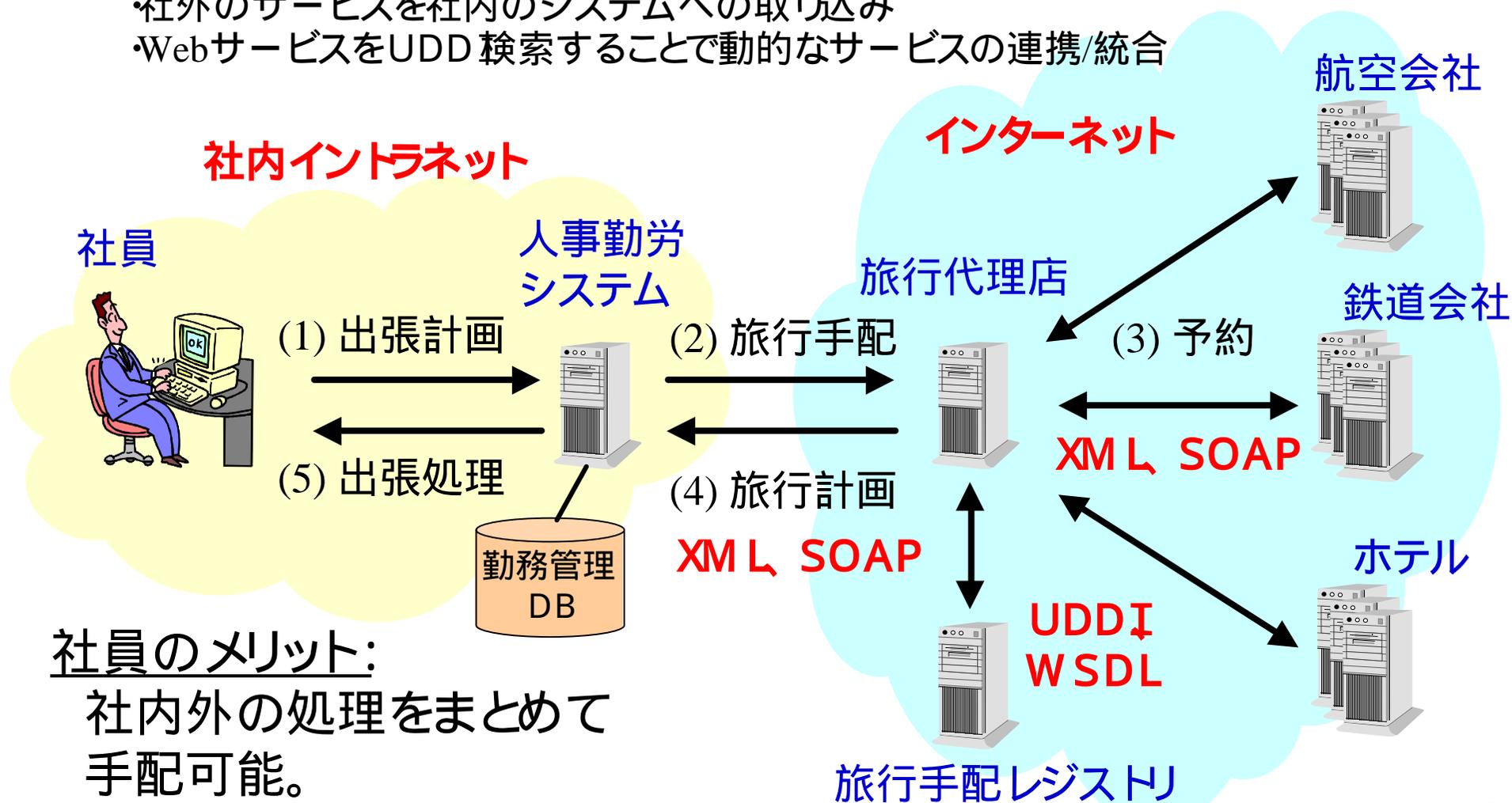
製品受注に対し、製品メーカーの生産計画を参照し、さらには部品メーカーの生産 / 納品計画も参照し、納期を回答。



Webサービス事例 (3) 総合出張サイト

社内のお出張処理システムの中に旅行代理店の旅行手配サービスを組み込んで提供。

- ・社外のサービスを社内のシステムへの取り込み
- ・WebサービスをUDD検索することで動的なサービスの連携/統合

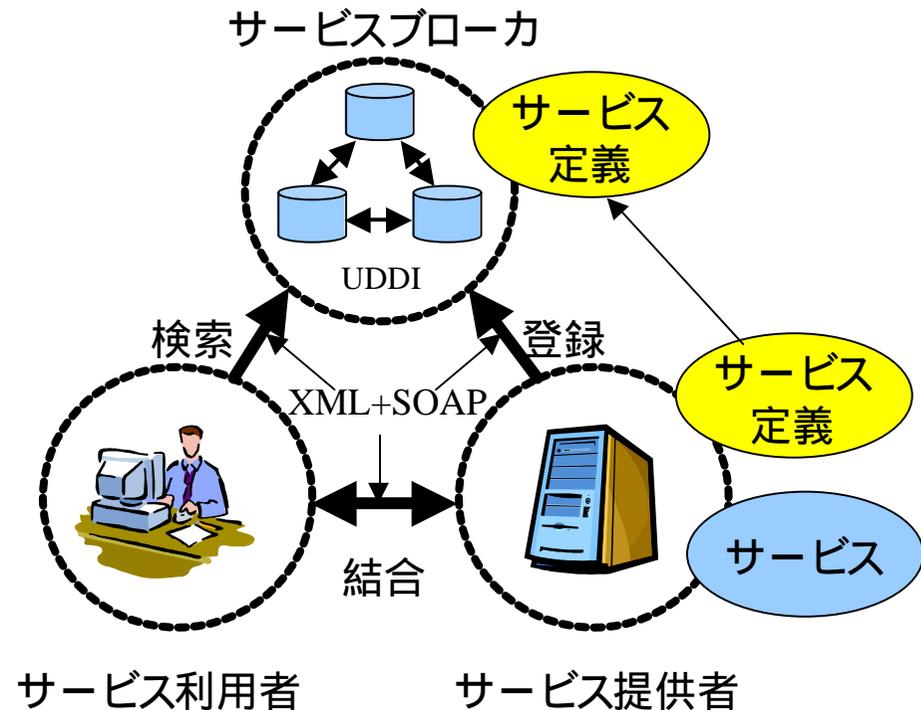


社員のメリット:
社内外の処理をまとめて
手配可能。

1.3 Webサービスの技術的特徴

- 「インターネット上で提供される疎結合型ソフトウェアコンポーネント」(ガートナーグループ)
 - インターネット標準のHTTPの上で、fire wall を越えてXMLによって相互に結合
 - サービスの実装(言語、ソフト、ハード)を隠蔽して、インタフェースの公開/検索/呼出による結合
 - インターネット上の既存アプリケーションを部品として利用することで、開発期間を短縮、費用を削減、コアコンピタンスへの集中を助長

疎結合なアクター(利用者、ブローカ、提供者)をつなぐためのサービスモデル



1.4 ビジネスへのインパクト

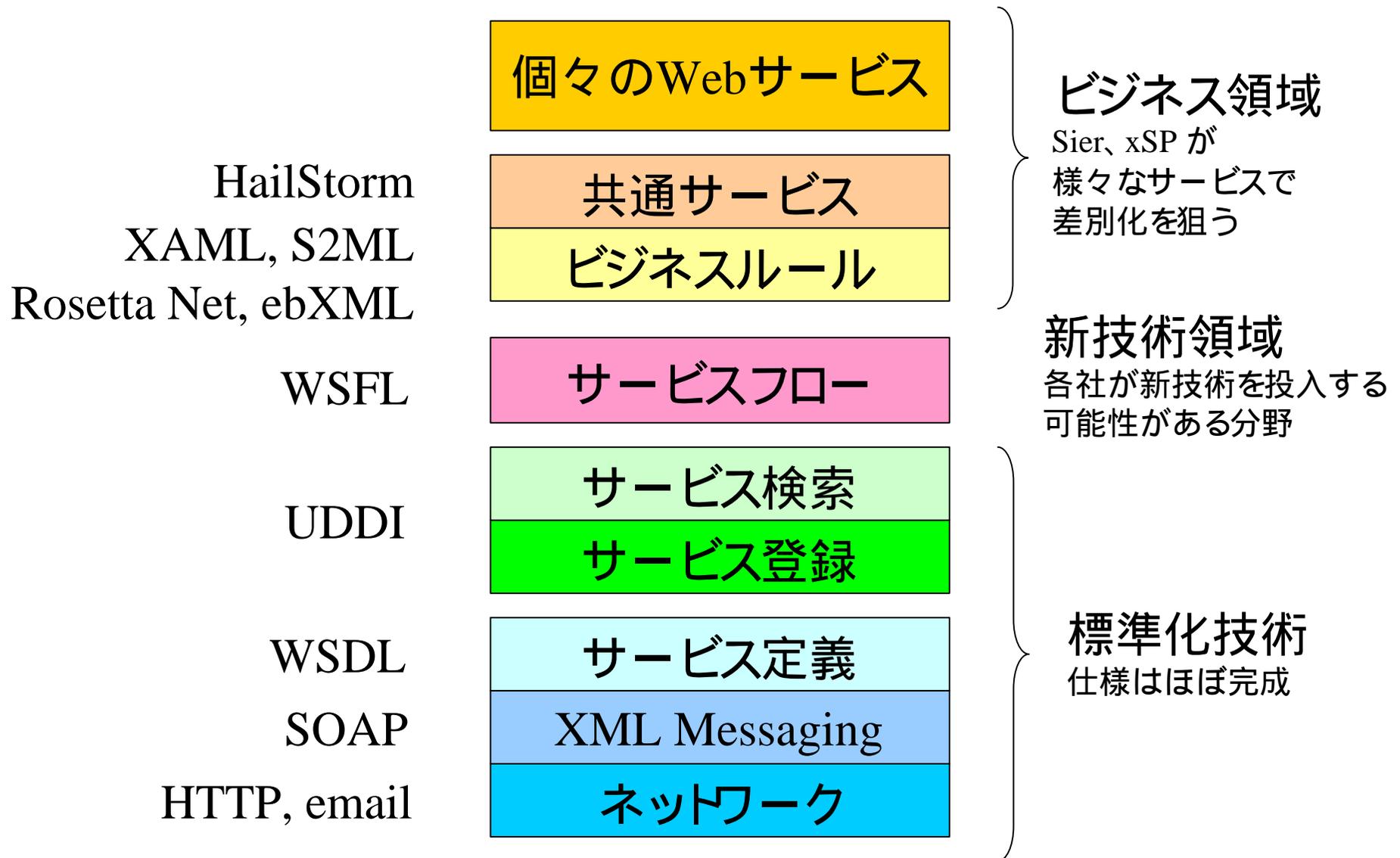
- パッケージ化したサービス提供ビジネスから、専門サービスの提供ビジネスと統合ビジネスに分化(decompose & integrate)
- ビジネスパートナーの組み替えを促進
- 新しいビジネスモデルを用いたビジネスへの新規参入を助長
- 顧客向けにパーソナライズ、状況に合わせたビジネスなど、ビジネスの柔軟性が必要とされる

2. Webサービスを支える技術

Webサービスを支える技術

- XML(eXtensible Markup Language)
 - タグを使って構造化された電子文書
- SOAP(Simple Object Access Protocol)
 - 分散システム間で構造化された型付き文書情報を交換するためのXMLベースのメッセージングプロトコル
 - アプリケーションやトランスポートプロトコルによらない情報交換のための標準
- WSDL(Web Services Description Language)
 - サービスの呼出手続きのインターフェースを規定するためのXMLフォーマット
- UDDI(Universal Description, Discovery and Integration)
 - Webサービスのための分散型情報レジストリの仕様
 - サービスの記述の登録と検索のための標準仕様
- WSFL(Web Services Flow Language)
 - 企業間、企業内のWebサービス統合に関するワークフローを定義する言語

2.1 Webサービススタック



XML とは

(eXtensible Markup Language)

インターネット時代の新しいドキュメント記述言語

[特徴]

- SGMLから受け継いだタグを用いた簡便な記述性
- HTMLから受け継いだシンプルな文法
- 人が見て利用する情報の表現手段に加えて、コンピュータ・アプリケーションとの連携を可能にする
- XML 上で新しい標準化が次々と進展
 - XML Schema: XML で書いた XML 文書定義
 - RDF: リソース定義フォーマット
 - XSLT: XML 文書の変換ルール
 - XML Signature: XML 文書への署名
 - S2ML: シングルサインオンを実現する規約

XML文書の例

XML宣言

```
<?xml version="1.0" encoding="Shift_JIS" ?>
```

文字符号化方式としてシフトJISを利用

```
<!DOCTYPE orderlist SYSTEM "http://www.nec.com/orderlist.dtd">
```

DTD宣言

(独自のDTDを使用を宣言)

```
<orderlist>
  <order priority="high">
    <date>2000/07/07</date>
    <item>PC-LM500J32D</item>
    <volume>12</volume>
  </order>
  <order priority="middle">
    <date>2000/09/03</date>
    <item>PC-LC600J34DR</item>
    <volume>8</volume>
  </order>
</orderlist>
```

文書本体

(XMLインスタンス)

2.2 SOAP

インターネットに分散したシステム間で構造化された型付きメッセージを交換するためのXMLの標準案。

RPC呼出

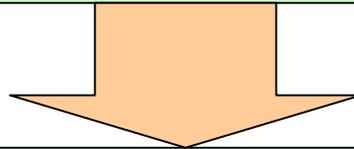
```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

RPC返り値

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse xmlns:m="Some-URI">
      <Price>34.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAPを使うことの利点

XMLで記述されたメッセージをHTTPやSMTP等の標準プロトコルを使ってメッセージを伝える



(1) インタフェースの違いを吸収できる

XMLの利点

DCOM、CORBA、Java™ RMIなどアプリケーションによって通信プロトコルが違う場合、SOAPを使うことによって連携が容易になる

(2) ファイアウォールを越えられる

HTTPの利点

企業で使用しているファイアウォールは、通常、DCOM、CORBA、Java™ RMIが使う通信プロトコルを通す設定にはなっていない

2.3 WSDL

WSDL(Web Services Description Language) は、サービスのエンドポイントで交わされる手続き呼出のインターフェースとプロトコルを規定するための XML フォーマット。

```
<?xml version="1.0"?>
<definitions name="StockQuote"
```

```
  targetNamespace="http://example.com/stockquote.wsdl"
  xmlns:tns="http://example.com/stockquote.wsdl"
  xmlns:xsd1="http://example.com/stockquote.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
```

```
<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/1999/XMLSchema">
    <element name="TradePriceRequest">
      <complexType>
        <all>
          <element name="tickerSymbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="TradePrice">
      <complexType>
        <all>
          <element name="price" type="float"/>
        </all>
      </complexType>
    </element>
  </schema>
</types>
```

データ型
の定義

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

メッセージ
の定義

```
<message name="GetLastTradePriceOutput">
  <part name="body" element="xsd1:TradePriceResult"/>
</message>
```

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

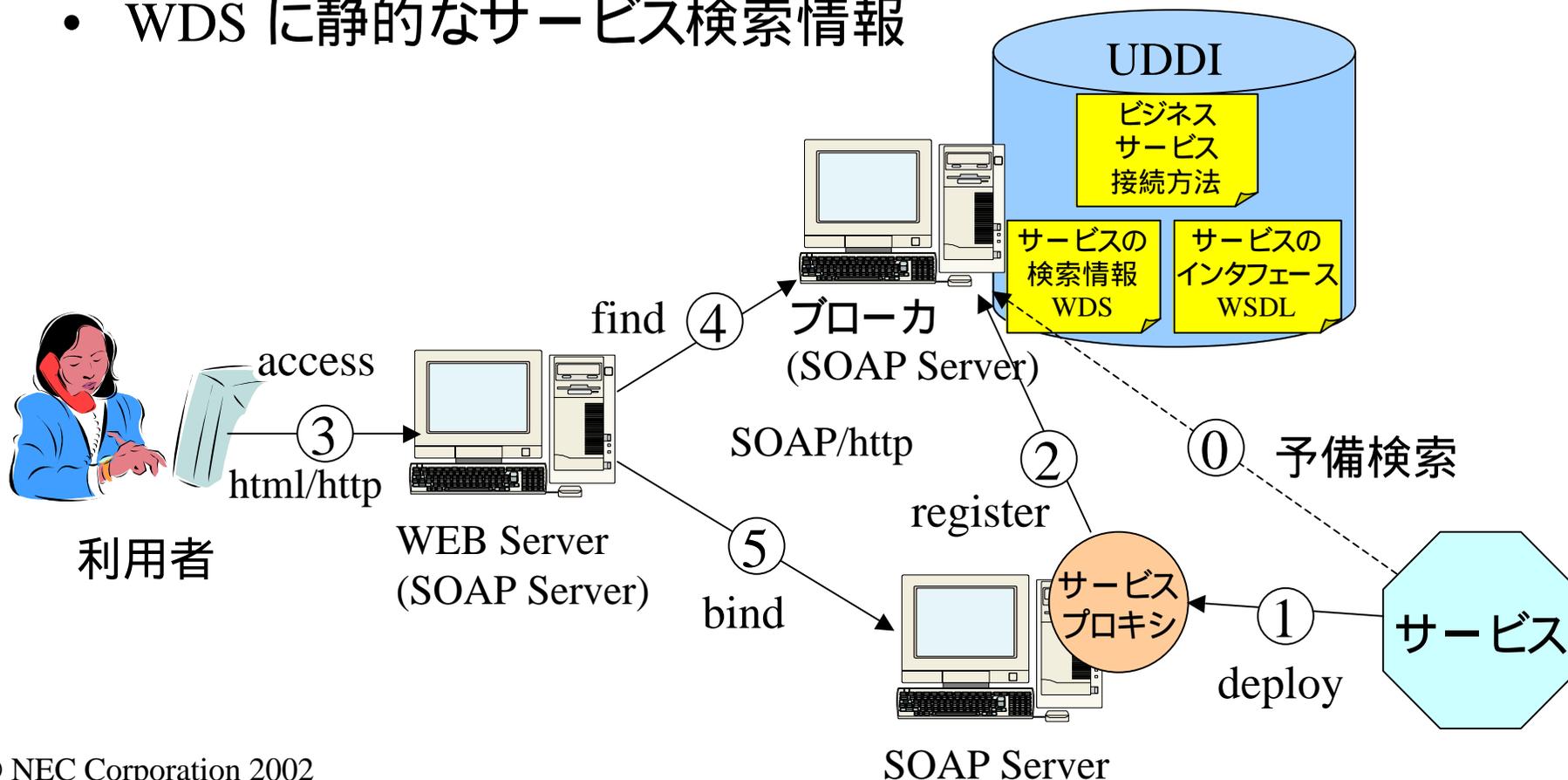
操作の
定義

```
<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" namespace="http://example.com/stockquote.xsd"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </input>
    <output>
      <soap:body use="literal" namespace="http://example.com/stockquote.xsd"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
    </output>
  </operation>
</binding>
```

プロトコル
バインディング

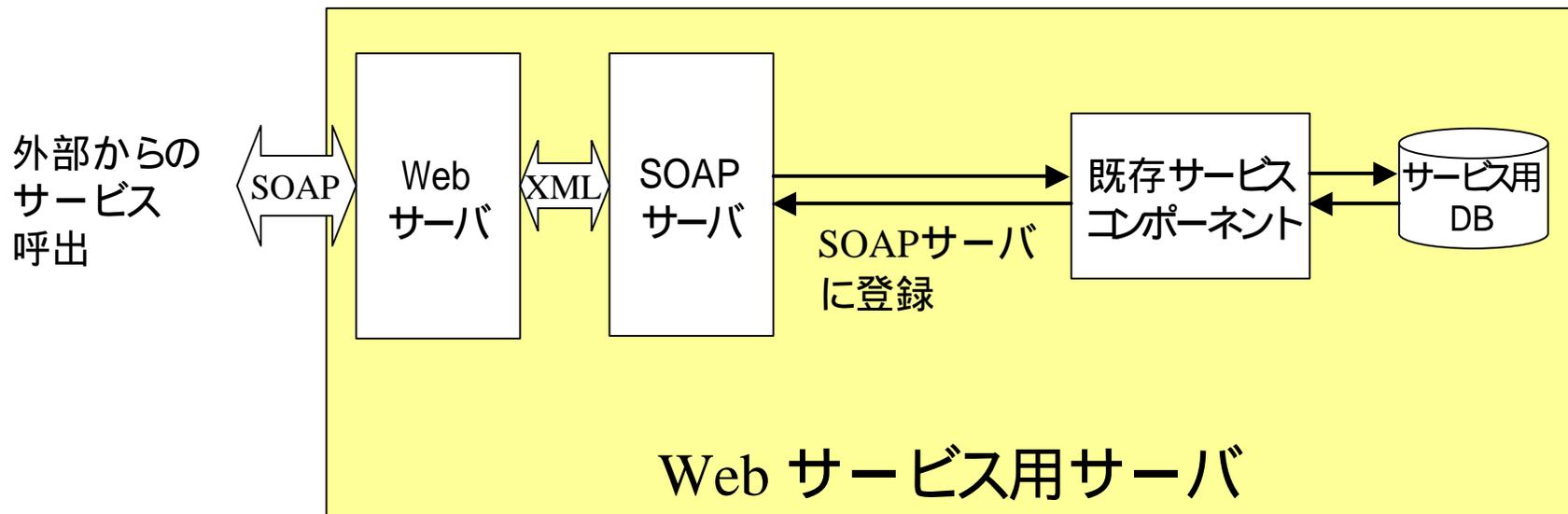
2.4 UDDIのアーキテクチャ

- SOAPサーバを連携させたアーキテクチャ
- UDDIの基本ドキュメントには、ビジネス(会社情報)/サービス/接続方法を記述
- WSDLに動的なサービス接続情報(インタフェース)の記述
- WDSに静的なサービス検索情報



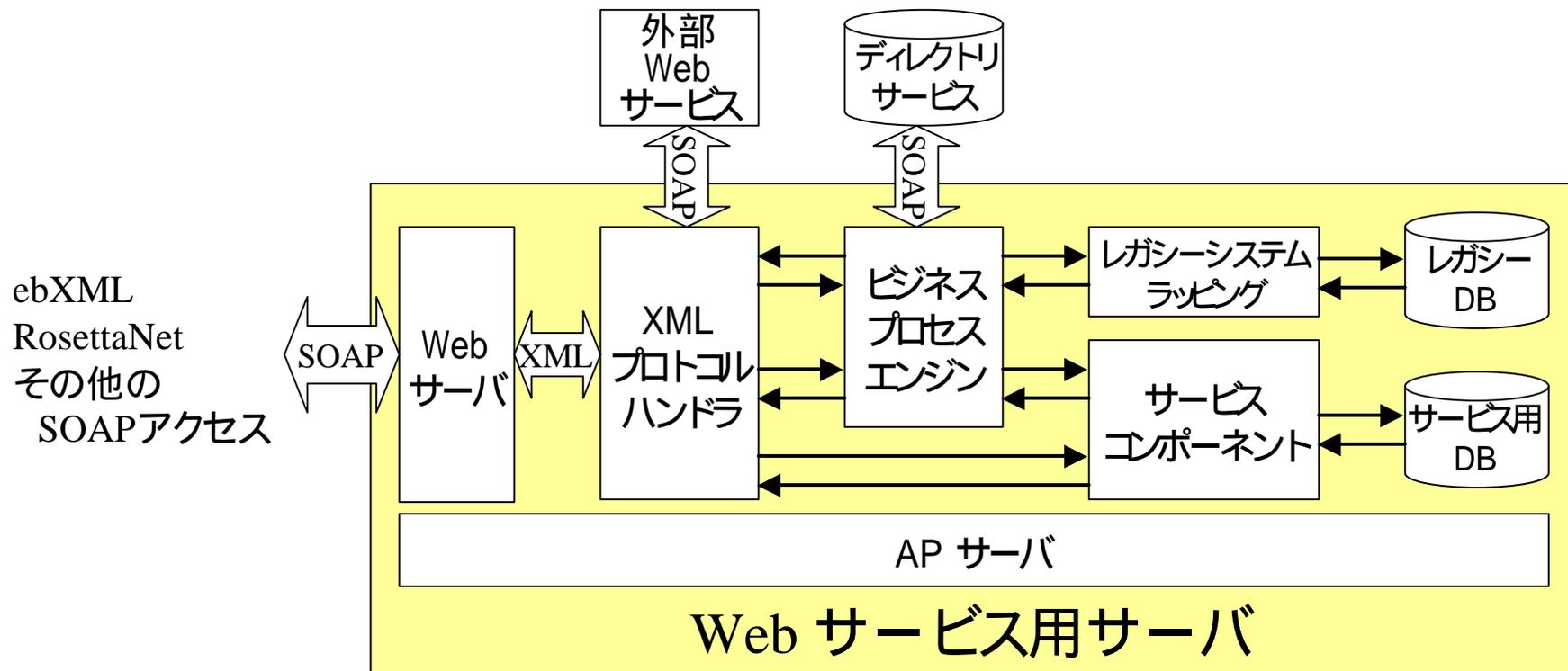
2.5 最小構成の Web サービス用サーバ

- これまで、Webに公開していたサービスを SOAP インタフェースで公開する。
 - 株価情報、天気予報のようなメソッド呼出だけで終わるサービスなら、すぐに Web サービス化が可能
 - 優良コンテンツがあれば、アクセス権によるビジネスが可能



2.6 高度なWebサービス向けアーキテクチャ

- SOAP で呼び出せるサービスを WSDL で定義し、UDDI に公開。
- ビジネスプロセスエンジンが、サービス統合を支援
 - 取引プロトコルを忠実に実行
 - サービスの一部を、外部の Web サービスに委託

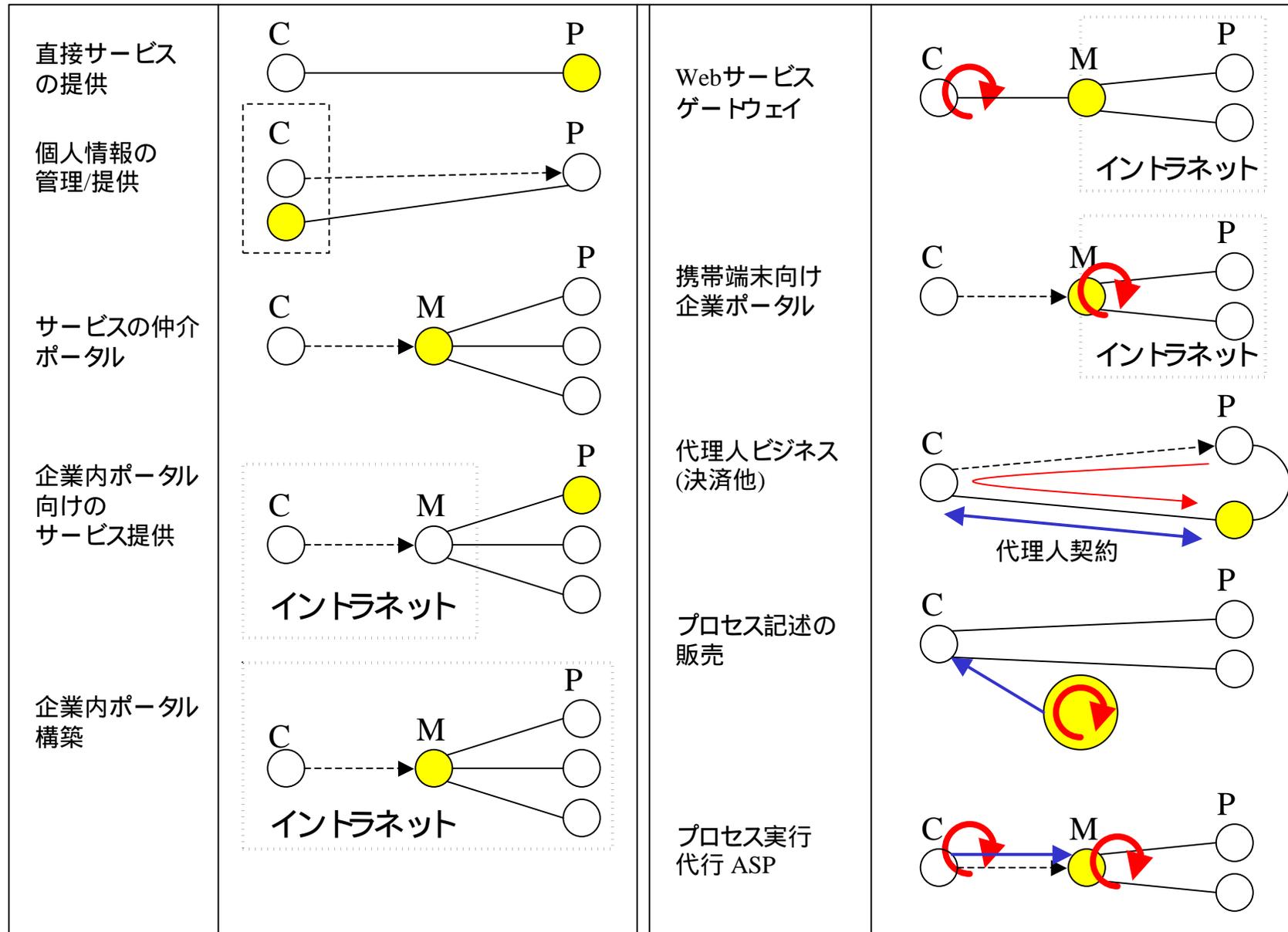


3. Webサービスのビジネス

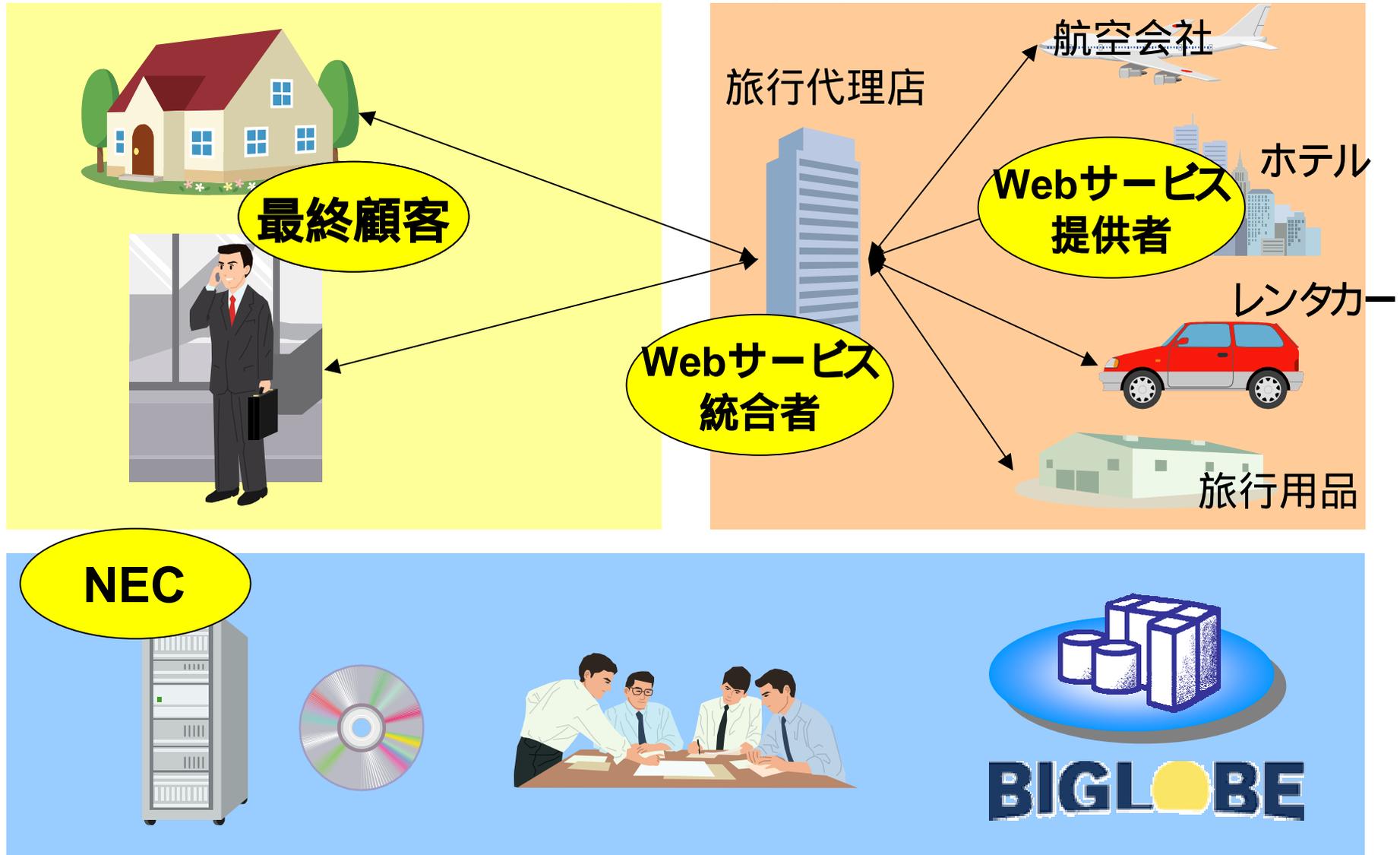
3.1 Web サービスのビジネス領域

ビジネス領域	特長
旅行業	<ul style="list-style-type: none"> • 鉄道、航空、ホテル、イベントなどのチケット予約/販売を行う複数のサービス提供企業を、インターネット上で統合的にサービス • 企業向けの出張予約システムのアウトソーシング
金融業	<ul style="list-style-type: none"> • パッケージ化された金融サービスから、顧客指向でサービスを組上げるカスタマイズ型の金融サービスへ。 • 複数の金融関連業者間をつなぎ、新たな金融業のバリューチェーン構築。
官公庁のオンラインサービス	<ul style="list-style-type: none"> • 複数官庁間にまたがる行政のワンストップサービスの実現方法に Web サービスを利用。 • 既存の Web ベースのオンライン行政サービスを、Web サービスベースのシステムに置き換え。
マーケットプレイス/EAI	<ul style="list-style-type: none"> • ブラウザ向けの e マーケットプレイス構築を脱し、企業間をつなぐ Web サービスへ。 • 企業内 EAI に、サービスのアウトソーシングを含めた新しいビジネスモデルが出現。
ビジネスプロセスの代行サーバの運用サービス	<ul style="list-style-type: none"> • 中小企業向けに、ビジネスプロセスの管理・運用サービスを代行するためのホスティングサービス。

3.2 Webサービスのビジネスパターン



3.3 Webサービスのメリット Webサービスのプレイヤー



Webサービスで期待されるメリット

- 最終顧客にとって
 - いつでも、どこでも様々なデバイス (携帯電話、PDA、PCなど) からサービスに接続できる (ユビキタス・コンピューティング)
 - ひとつずつページのリンクを辿るのではなく、最適なサービスを自動的に検索してくれる
 - ワンストップ・シングルログオンで複数サービスを受けられる
 - 個人向けにカスタマイズされたサービスを受けられる
- Webサービス統合者にとって
 - 自社に無い機能を外部機関にアウトソースできる
 - ワンストップサービスの提供が容易になる
 - 取引先のバリエーションが増え、サービスの魅力が増す
 - 異業種による新しいパートナー関係から新しいビジネスが生まれる
- Webサービス提供者にとって
 - 自社の得意な領域に資源を集中できる
 - レジストリを通じて、新しい取引相手を見つけることができる

Webサービスで期待されるメリット (開発者の視点から)

- Webサービス統合サイトの開発にとって
 - ポータルサイトの構築が容易になる
 - 処理の自動化を行うのが容易
- Webサービス提供サイトの開発にとって
 - プロトコルがオープンなので、インターネット、イントラネットを問わず、様々なプラットフォーム間での連携が可能
 - システム連携の開発コスト削減が見込める
 - データはXMLなのでシステムの変更が容易

4. Webサービスに対する業界動向

4.1 北米ベンダー動向

- HP
 - 1999年頃から、e-services というキーワードで、インターネットのビジネス統合のビジョンを打ち出す。
- Microsoft®
 - Windows® の後継として、.NET の構想を打ち出す。ネットワークからのソフトの提供を含めて、Windows® のインターネット版を狙う。C#, CLR による統合を
- IBM®
 - Web Services Architecture という標語の下に、これからのシステムインテグレーションのプラットフォームを提案
- Sun™
 - Microsoft® に対抗して、Java™ によるソリューション構築を主張。

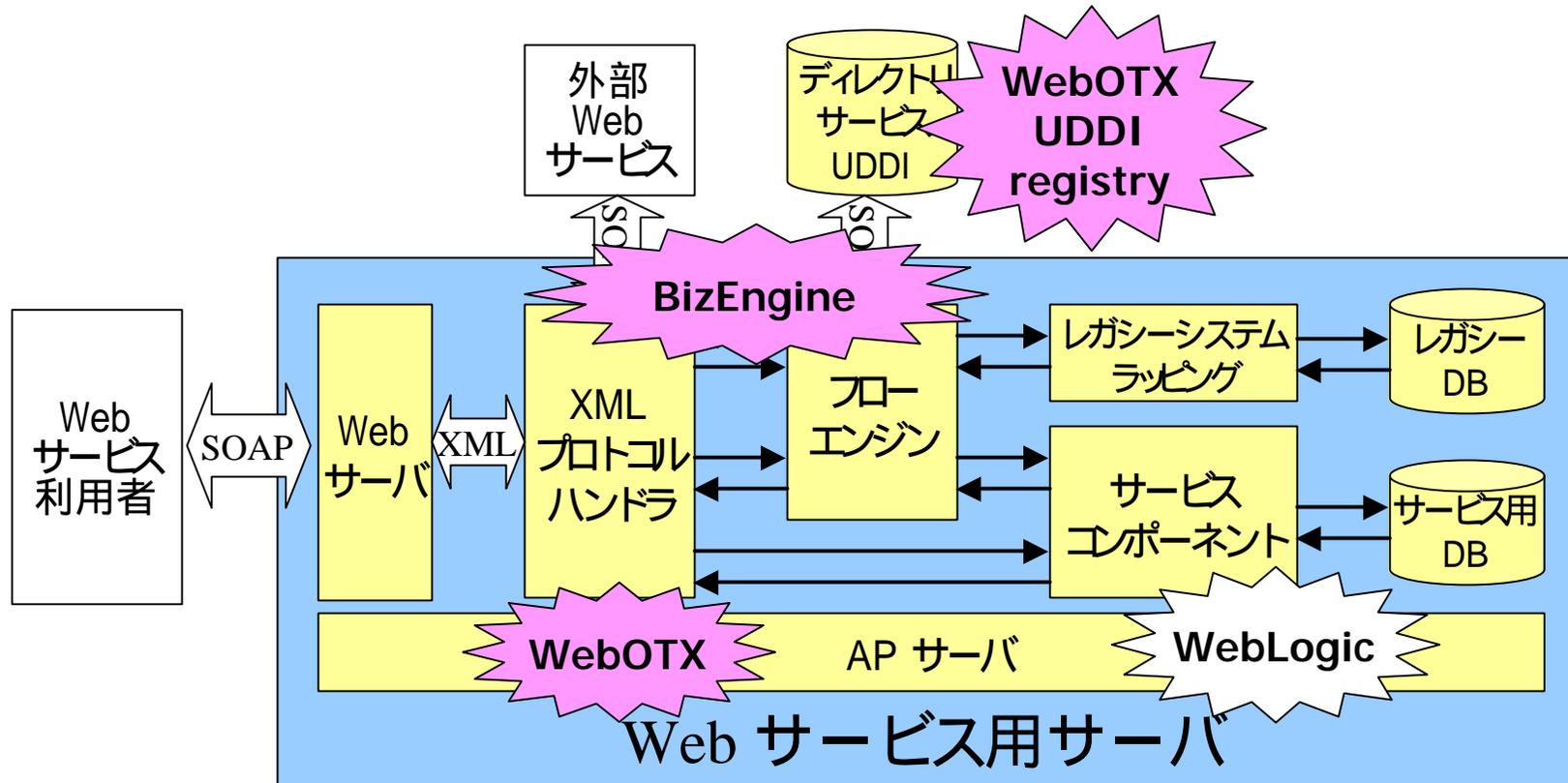
4.2 標準化動向

- W3C
 - SOAP ベースに、XML Protocol の規格を検討中
 - WSDL をNoteとして受け付け (2001/3)
- ebXML
 - UN/CEFACT, OASIS が承認 (2001/5)
 - メッセージング(SOAP)、レジストリ/レポジトリから、ビジネスプロセス定義スキーマやパートナー間契約文書まで、幅広い標準化を推進
- UDDI
 - ビジネスディレクトリのアクセスインタフェースを標準化 (第2版 2001/6)

5. NECの取り組み

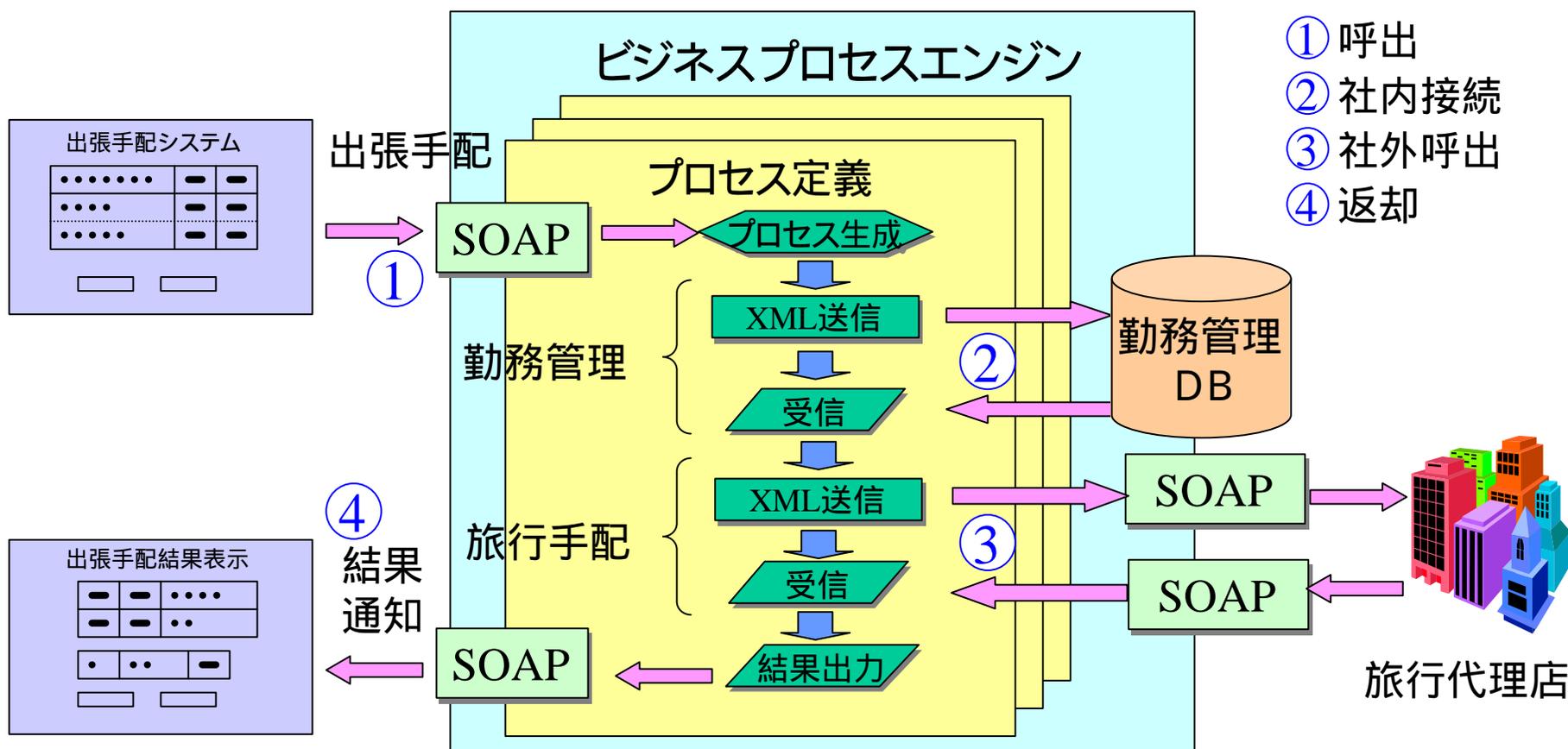
NECのWebサービス製品群

- Web/AP サーバ: WebOTX, WebLogic Server
- ビジネスフローエンジン: BizEngine/Base
- XMLプロトコルハンドラ: BizEngine/WS
- UDDI: WebOTX UDDI Registry



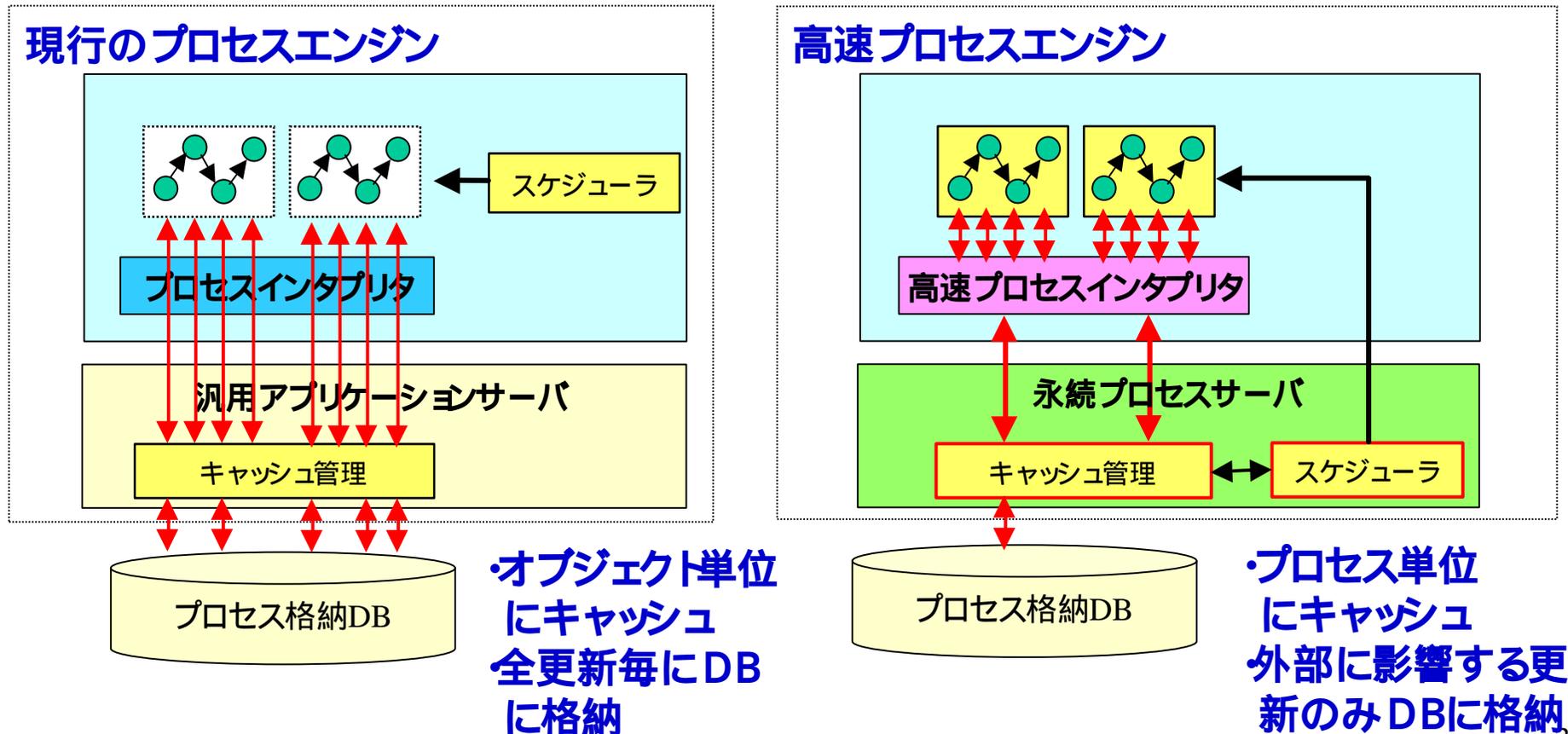
5.1 ビジネスフローエンジン

- XML文書でサービス・リクエストを受付
- 社内システム呼出し、他のWebサービス呼出し、サービスの組合せ等をプロセス定義に則って実行
- XML文書のサービス結果を返却



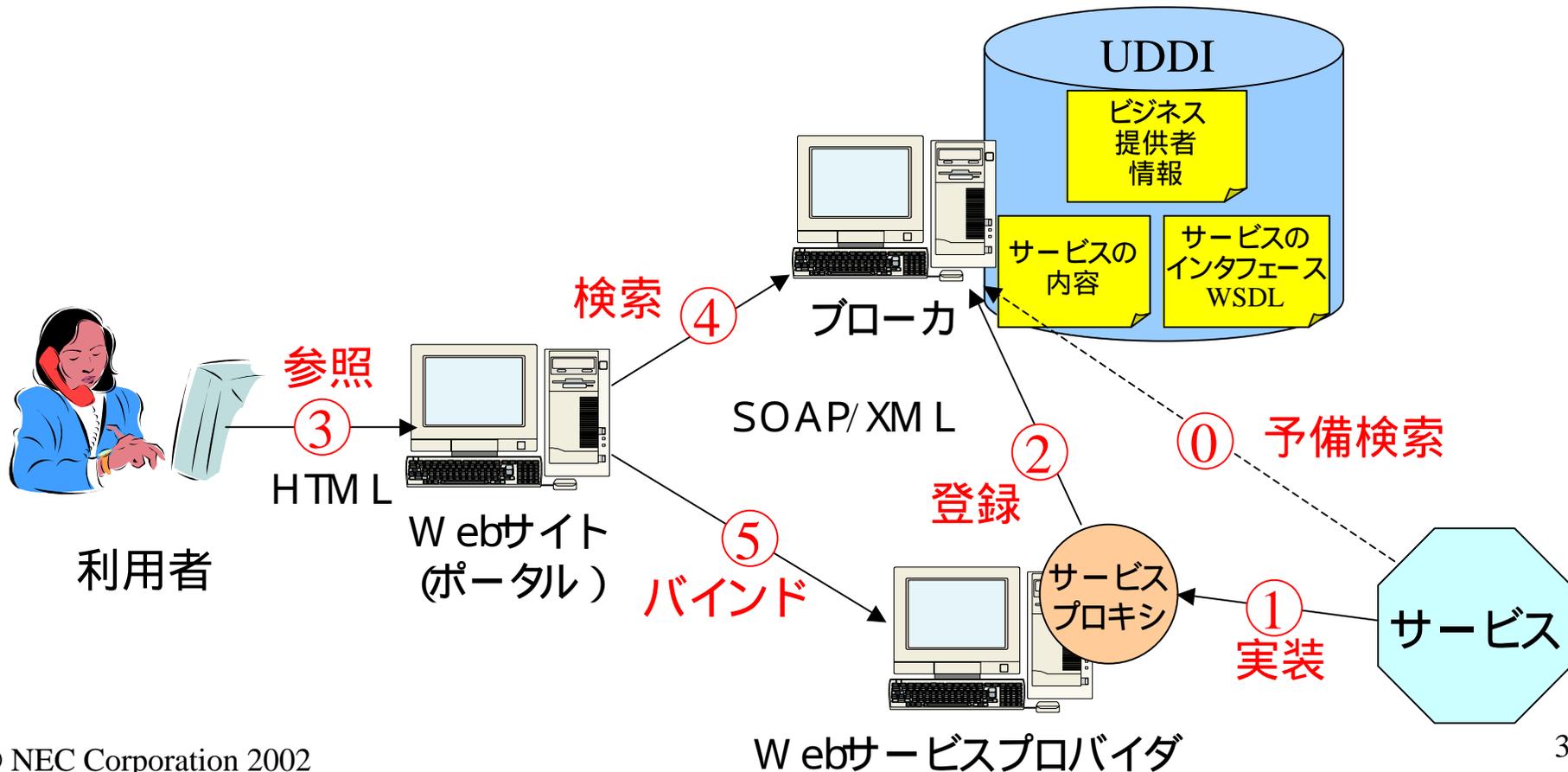
BizEngine/Base の特長

- ・プロセス単位のキャッシュでDBアクセスを効率化
研究所のエージェントサーバ技術を活用
- ・プロセスインタプリタで不要なDBアクセスをブロック
外部に影響を与えない更新はキャッシュのみ更新



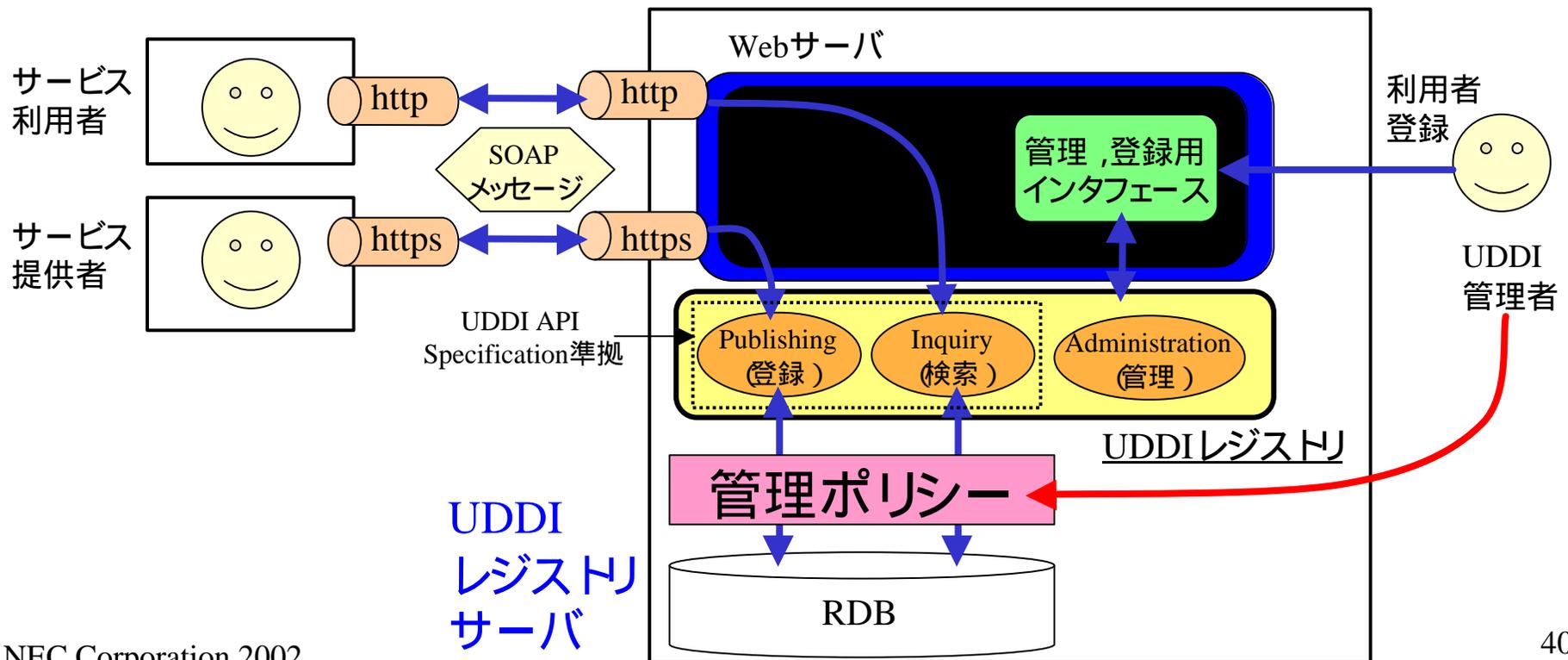
5.2 UDDIレジストリ

- ビジネス/サービス/接続方法に関する文書を登録し、利用者に検索機能を提供
 - 登録、検索のインターフェースもSOAP から利用できる。
- 動的なWebサービスの選択/統合を可能に



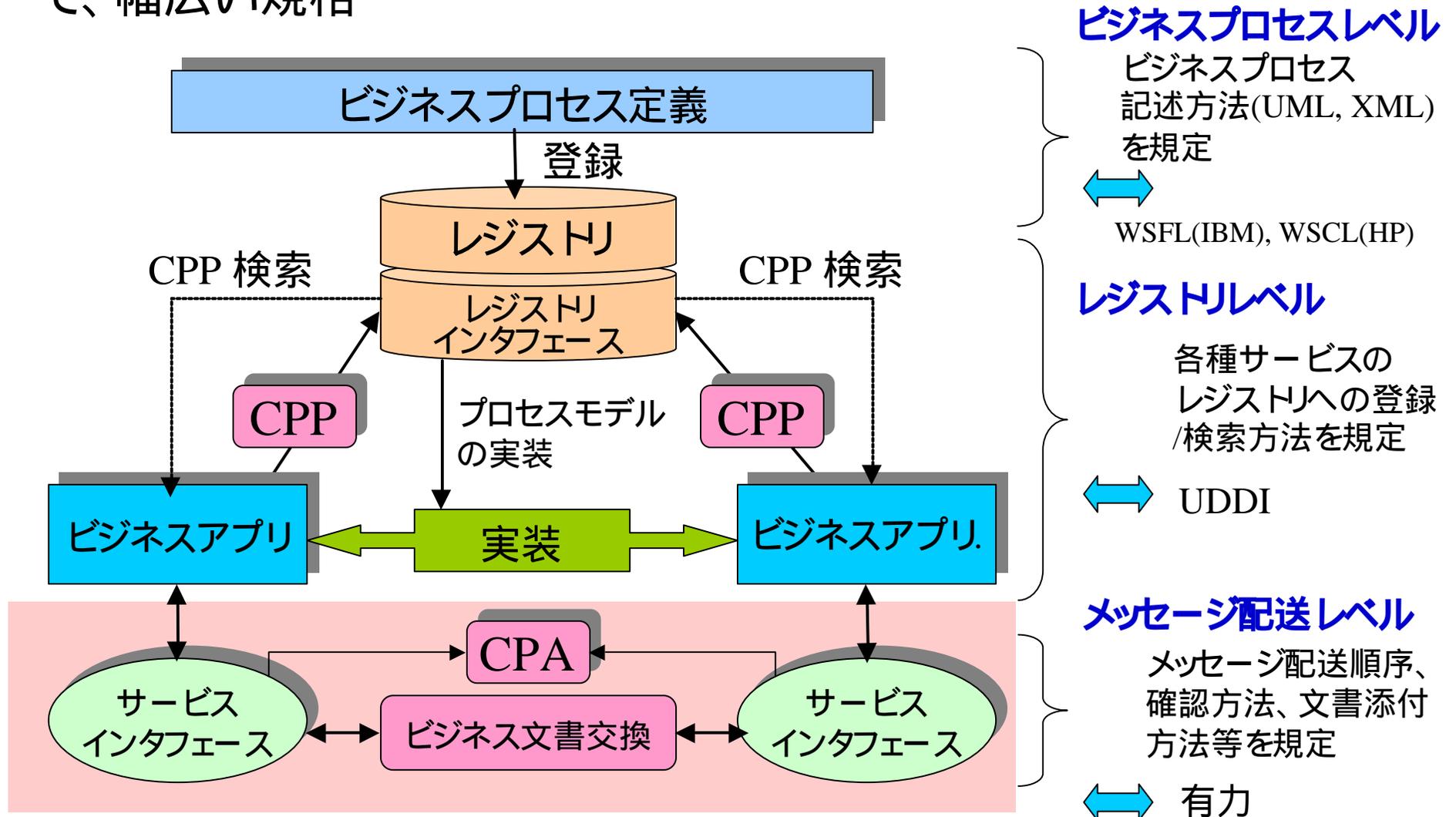
WebOTX/UDDIの特長

- UDD IV.2.0 (2001年6月承認)に準拠
- UDD IXML 文書の蓄積/検索をRDB にマップする技術を確立。



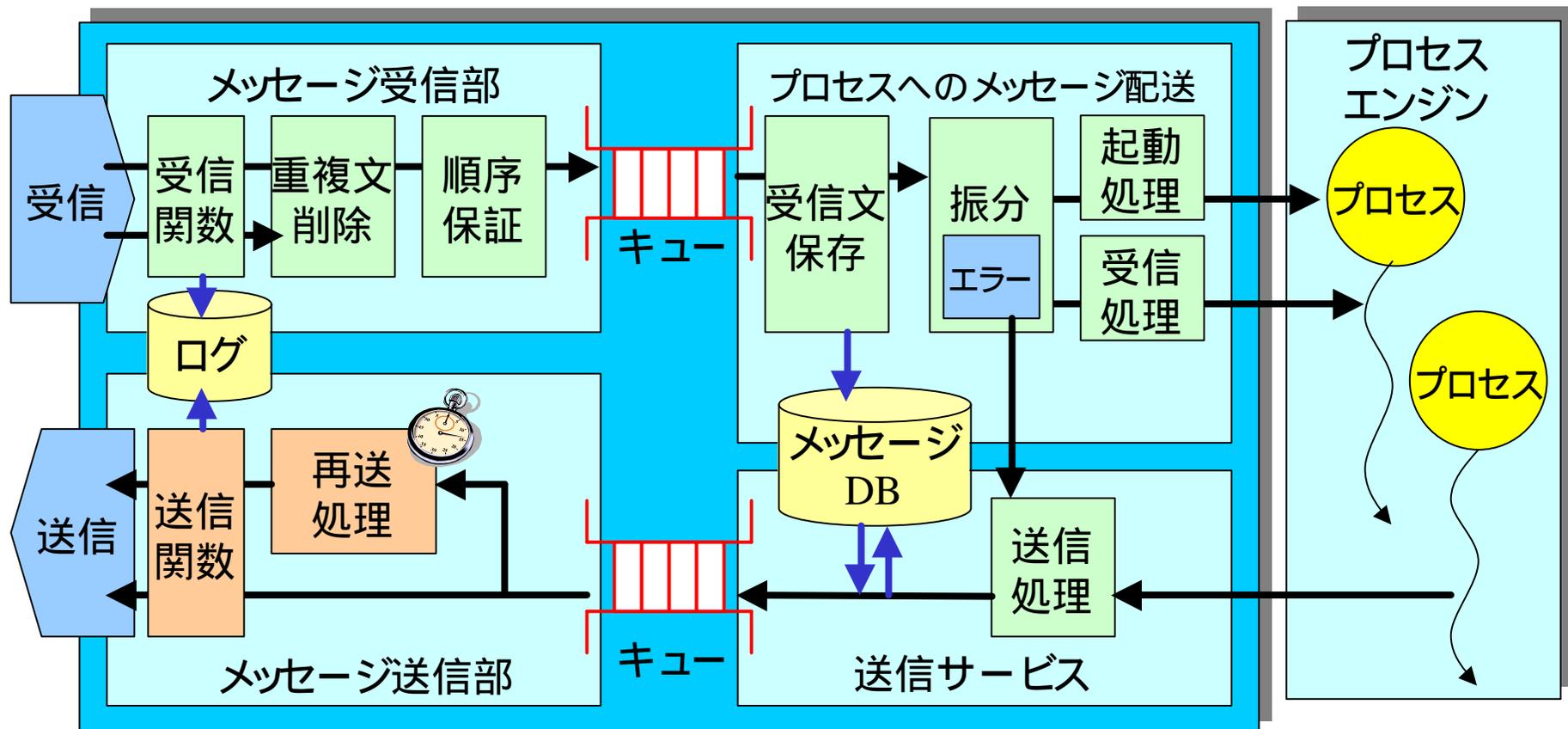
5.3 ebXML Lプロトコルハンドラ

ビジネスプロセス定義スキーマから、レジストリ、メッセージ配送まで、幅広い規格



BizEngine/WSの特長

- ebXML の高信頼メッセージサービス仕様を完全サポート
- マルチスレッドで高スループット、デッドロックフリーな設計。
- 障害発生時に障害時点からの再起動可能。



6. Webサービス発展のステップ

Webサービス発展のステップ

	Step 1	Step 2	Step 3
	シンプルWebサービス	Webサービス統合	動的Webサービス統合
適用技術	XML, SOAP	グループ内UDDI, WSDL ebXML, RosettaNet, .NET, HailStorm	UDDI, WSFL, UBL
特徴	<ul style="list-style-type: none"> •F/W 越えしたサービス呼出 •異なる計算機上のサービスを同一インタフェースで統合 	<ul style="list-style-type: none"> •グループ企業内ビジネスディレクトリの運営 •ワークフロー定義に基づくサービス統合 •RosettaNet 等の先行ビジネスプロセスの Web サービス化 	<ul style="list-style-type: none"> •パートナーのダイナミックな検索 •オンラインのビジネス契約に基づく取引 •企業間ワークフローの設計と実行、再利用
ビジネス領域	<ul style="list-style-type: none"> •強い単発のサービスをWebサービス化して提供 •サービスのアウトソーシングによるビジネスの補強 •inB 領域における社内情報システムの統合 	<ul style="list-style-type: none"> •Webサービスを用いたB2B の固定取引が本格化 •複数サービスのアグリゲーションを行うASPビジネスの台頭 	<ul style="list-style-type: none"> •ビジネスパートナーとの動的なバリューチェーン構築が本格化 •サービス品質を保証する保険サービス、高信頼レジストリの出現

7 .Webサービスの課題

Webサービス・ビジネスの課題

- Webサービス利用者にとって
 - 個別サービス品質の見極め責任を負うことになる
- Webサービス・アグリゲータにとって
 - 統合する個々のWebサービスの品質確認が必要
 - 最終顧客に保証するに足るサービスを提供してもらえるか
 - 最終顧客にサービスを提供する時に、そのサービスが確実に利用できるか
- Webサービス提供者にとって
 - 競争が激化し、良いコンテンツを持つものだけが生き残る
 - 価格競争に拍車がかかる

Webサービス・システム開発者の課題

- XML文書処理は、従来の通信プロトコルよりもデータ量、処理量が増大
 - パフォーマンスの低下への対策
 - XML文書のデータベースへの格納方法
- サービス統合を正しく行えることの検証の必要性
 - 取引プロトコルどうしの整合性の検証
 - 設計時のプロトコルの整合
 - 実行時のプロトコルの整合
- 動的な分散システム設計開発におけるデバッグや実行同期などの開発作業

8. まとめ

- Webサービスに対する活動が非常に活発
 - 米国企業の製品発表
 - 標準化動向(SOAP, UDDI, WSDL, ebXML)
- 「見る」Webから、「使う」Webへ
 - ビジネスを対象にしたインターネット技術
 - サービスの組合せで、システム構築
- ビジネスへのインパクト
 - ビジネスパートナーの入れ替えが容易に
 - 実行時の状況で、サービスの選択が可能
 - 全てのサービスを持たない企業でも、社外のサービスを統合して、顧客に提供することが可能
 - 新しいビジネスモデルの登場

登録商標

- Sun™, Java™, Solaris™は、米国Sun Microsystems の商標あるいは登録商標です。
- Microsoft®, Windows®, Visual Studio™, BizTalk™, Sharepoint™ は、米国 Microsoft の商標あるいは登録商標です。
- IBM®は、米国 IBM の登録商標です。
- BEA WebLogic® は、米国 BEA Systems の登録商標です。