

【TravelXML利用Webサービス実証実験プロジェクト成果資料】



# TravelXMLを利用した Webサービス実証実験デモ デモ : 非同期システム連携

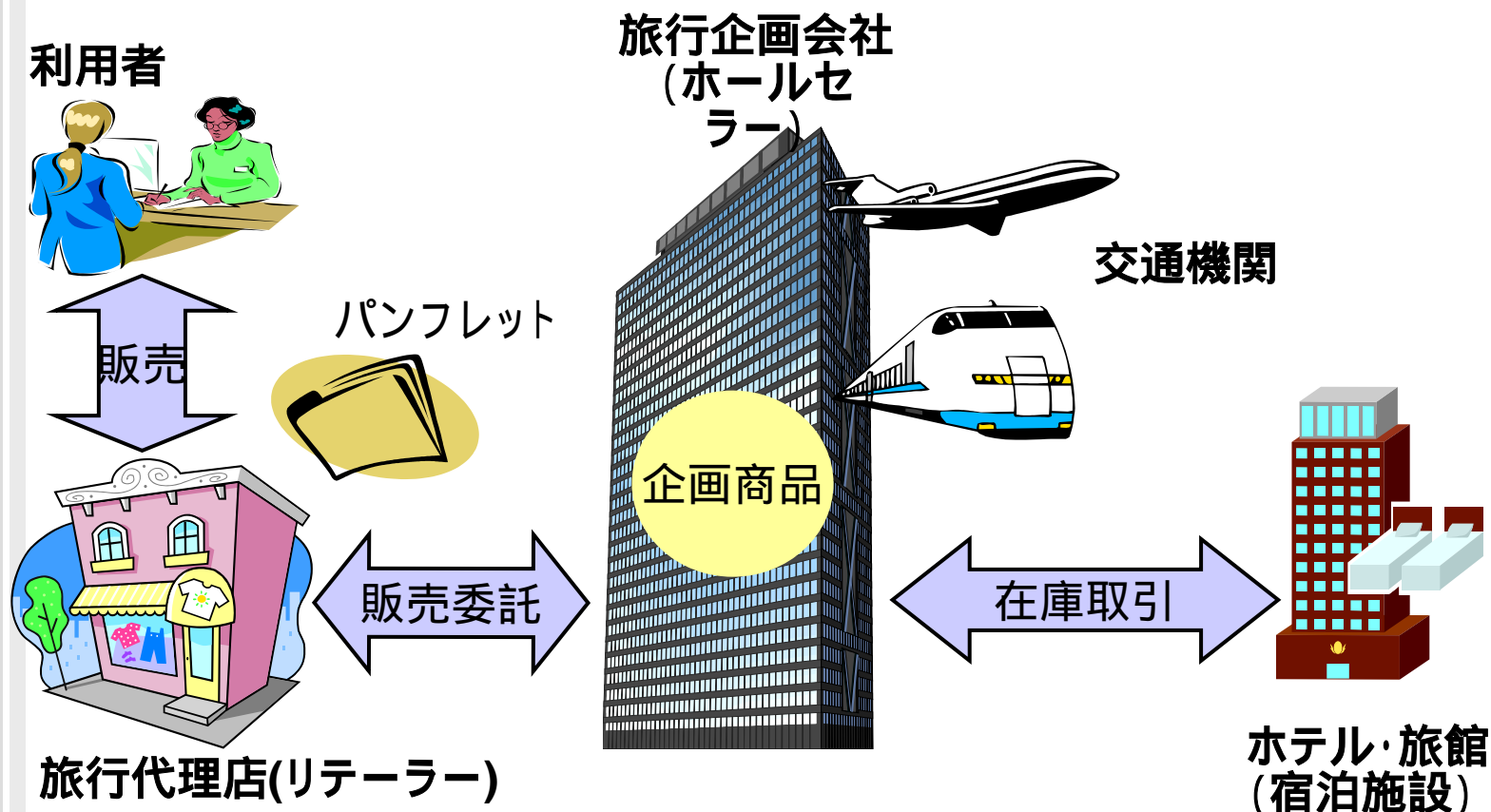
株式会社日立システムアンドサービス 村垣 委久夫  
赤澤 一憲  
株式会社ネット・タイム 本田 直紀  
日本ユニシス株式会社 尾島 良司

# アジェンダ

- 実証実験概要
- デモ 非同期システム連携
- デモ 参加企業
- デモシナリオ
- 旅行代理店システム詳細
- 旅行企画会社システム詳細
- ホテル・旅館システム詳細

# 実証実験の概要

- 日本特有のパッケージ旅行の取引をモデルに TravelXMLとWebサービス技術を利用した実証実験を行う。



# デモ 非同期システム連携



- なぜ非同期連携が必要なのか。
- 全てのケースを想定して自動化するのは大変。
- 小規模のホテルでは部屋在庫は貴重。状況を判断して遣り繰りしたい。
- 在庫を返すかどうか、人の権限で決めたい状況がある。

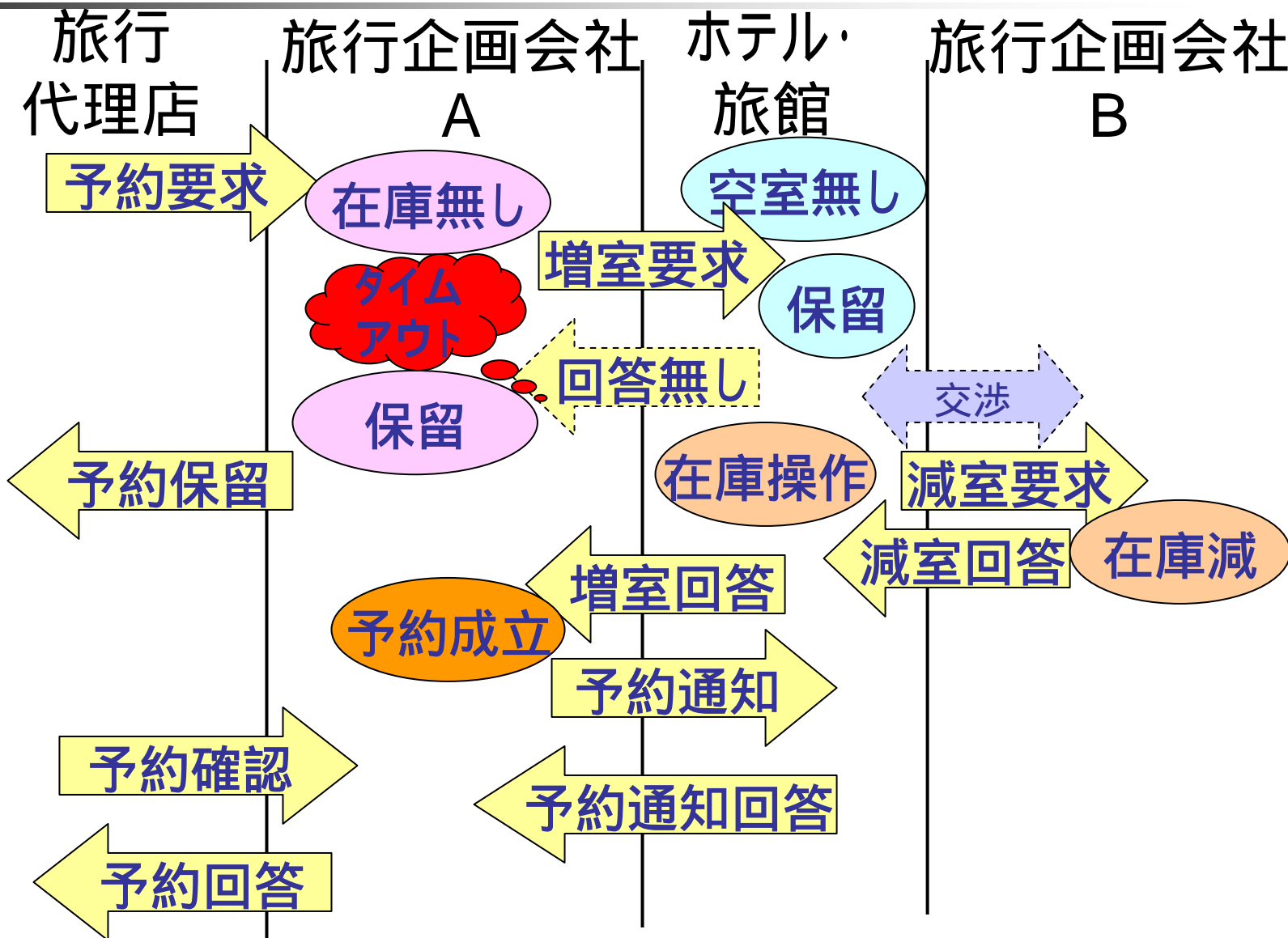
それなら

Webサービスによるシステム自動連係に人の操作を伴う処理を含めて、非同期処理を行うWebサービス実装間の接続検証を行う。

# デモシナリオ

- ホテル・旅館で旅行企画会社から増室要求を受けたが割り当てる空き室がない。他の旅行企画会社から返して貰いたい交渉に時間が掛かる。
  - 増室要求に応える空室が無い場合に一旦保留の回答を返す。
  - 返して貰えそうな旅行企画会社と交渉し、了解を得て在庫増減室操作により保留した増室要求に在庫割当を行う。
  - 旅行企画会社は予約確定により予約通知を行う。

# 想定したシーケンス



XML Consortium

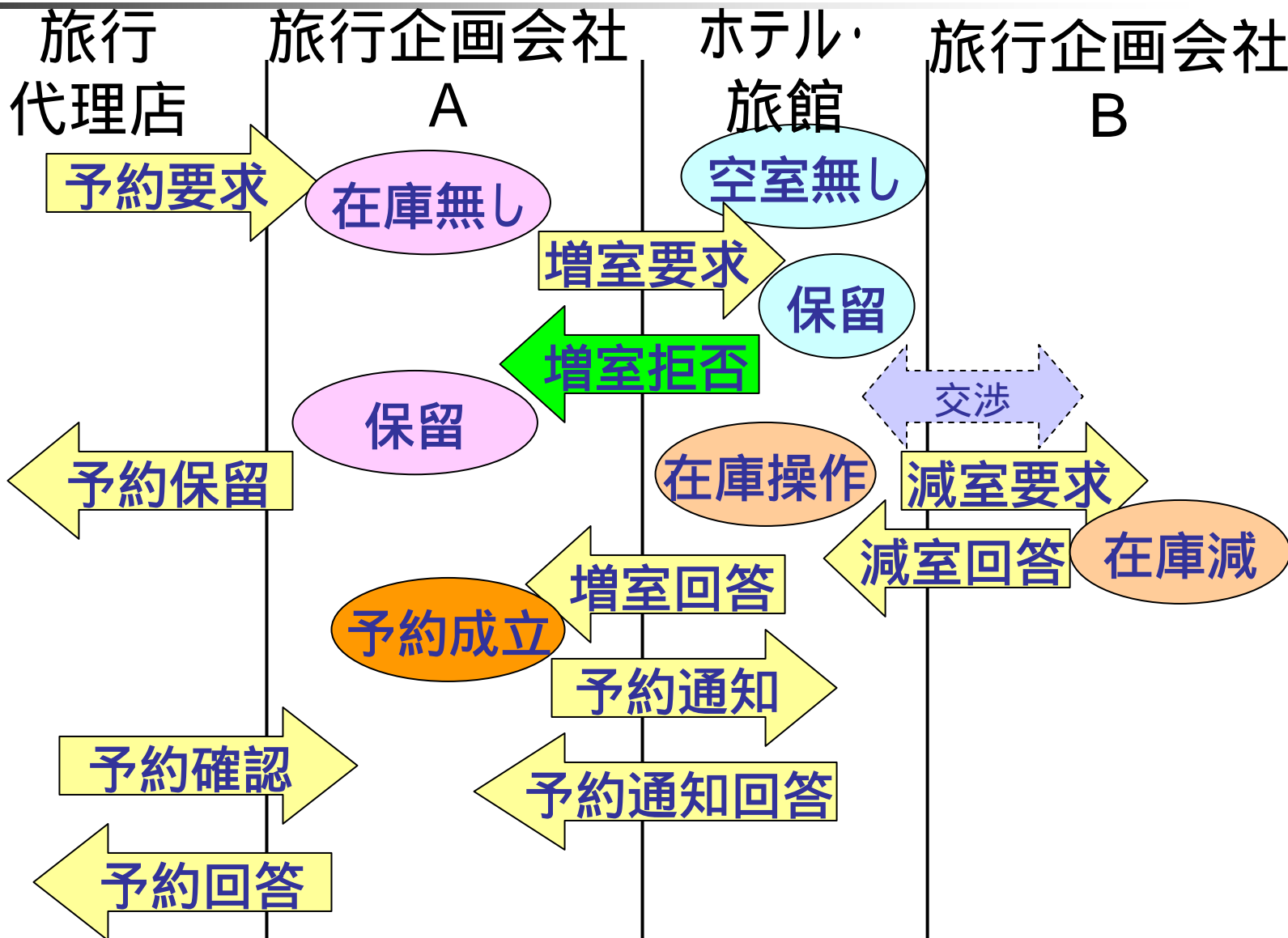
# 非同期の問題点

- 応答を監視し、応答がない場合にタイムアウトを検出した場合にアプリケーションで処理を行う実装が非常に難しい。
- 環境によりタイムアウトを監視する時間の設定が難しい。

そこで

- タイムアウトの検出に頼らずに応答を返す事にするが業務アプリケーションで非同期処理する実装とする。

# デモのシーケンス



XML Consortium



# デモ 参加メンバー企業



<b>役割</b>	<b>実装企業</b>	<b>UDDIレジストリ</b>
UDDI	株式会社 ブレイネットワークス	WebSphere UDDI Registry

<b>役割</b>	<b>実装企業</b>	<b>利用Webサービス実装</b>	<b>利用WS-Security実装</b>
旅行代理店	株式会社 ネット・タイム	AXIS(Client)	日立WS-Security実装
旅行企画会社	日本ユニシス株式会社	Microsoft BizTalk Server 2004	
	株式会社 日立製作所	Cosminexus Application Server Version6	日立WS-Security実装
ホテル・旅館	株式会社 日立システムアンドサービス 日本オラクル株式会社	PolarLake Messaging Integrator 3.2.6	日立WS-Security実装
	東芝ソリューション株式会社	Tomcat 4.1.30	日立WS-Security実装

<b>役割</b>	<b>実装企業</b>	<b>SOAPMonitor</b>
モニタ	PFUアクティブラボ株式会社	オリジナルSOAPMonitor 実装言語: Java GUI: Swing

<b>役割</b>	<b>実装企業</b>	<b>セキュリティ製品 / 提供API</b>
セキュリティ	株式会社 日立製作所	Apache XML SecurityベースWS-Security デモ用ライブラリ



XML Consortium

A decorative graphic on the left side of the slide, consisting of a vertical black line and a horizontal black line intersecting at a point. The background behind the lines is a gradient of purple and green.

## 旅行代理店システム

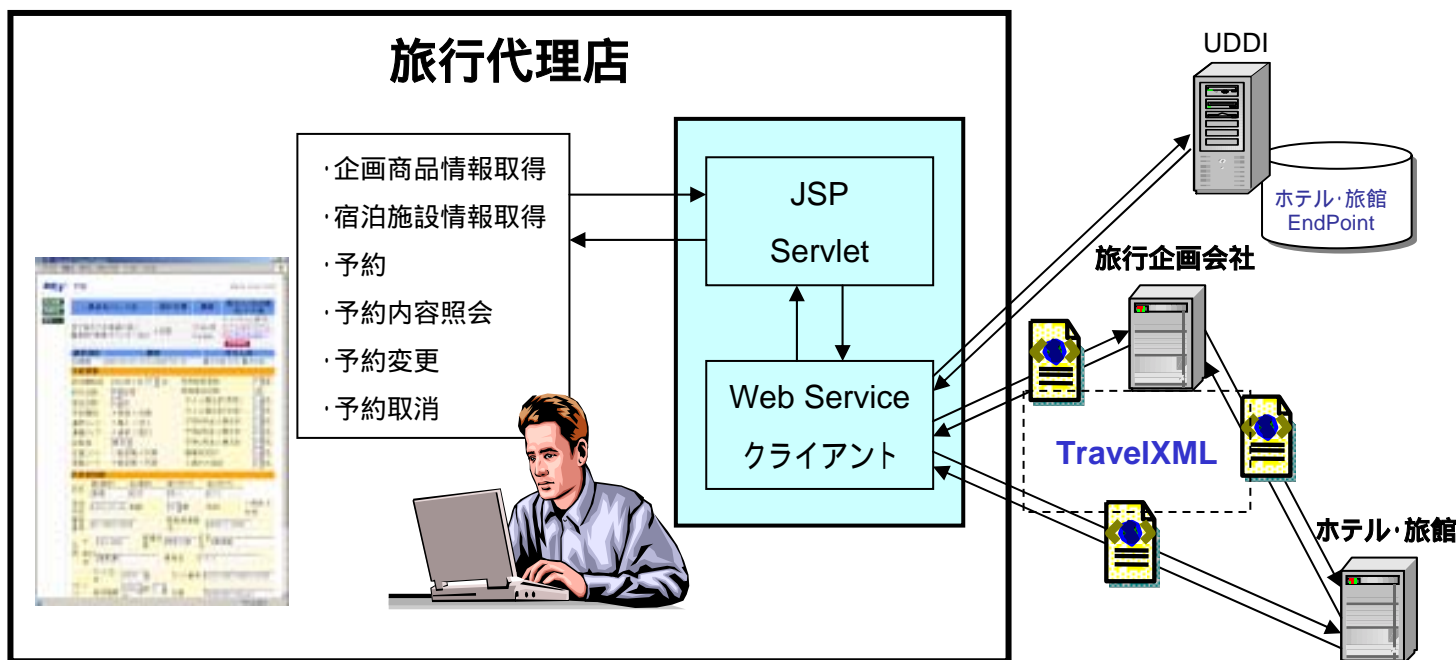
株式会社ネット・タイム 本田 直紀

# 概要

- **企画商品情報取得**
  - 検索条件(エリア、テーマなど)を入力し、企画商品情報を複数の旅行企画会社から取得する
- **ホテル・旅館情報取得**
  - 取得した企画商品情報をもとに、ホテル・旅館情報をホテル・旅館から取得する
- **予約**
  - 自分の証明書(秘密鍵)でセキュリティ情報に署名する
  - 利用するホテル・旅館の証明書(公開鍵)でセキュリティ情報を暗号化する
  - 取得した企画商品情報をもとに、予約情報を入力し、旅行企画会社に送信する
- **予約内容照会**
  - 取得した予約結果をもとに、予約情報を旅行企画会社から取得する
- **予約変更**
  - 取得した予約情報をもとに、予約変更を旅行企画会社に送信する
- **予約取消**
  - 取得した予約情報をもとに、予約取消を旅行企画会社に送信する
- **開発工数** : 約 1.0人月
  - セキュリティに関する部分:0.1人月
  - ロジックに関する部分0.5人月
  - GUI(画面)に関する部分:0.4人月
- **ステップ数** : 約 4.5 k ステップ(コメント除く)
  - セキュリティに関する部分:0.3 k ステップ
  - ロジックに関する部分:2.5k ステップ
  - GUI(画面)に関する部分:1.7k ステップ

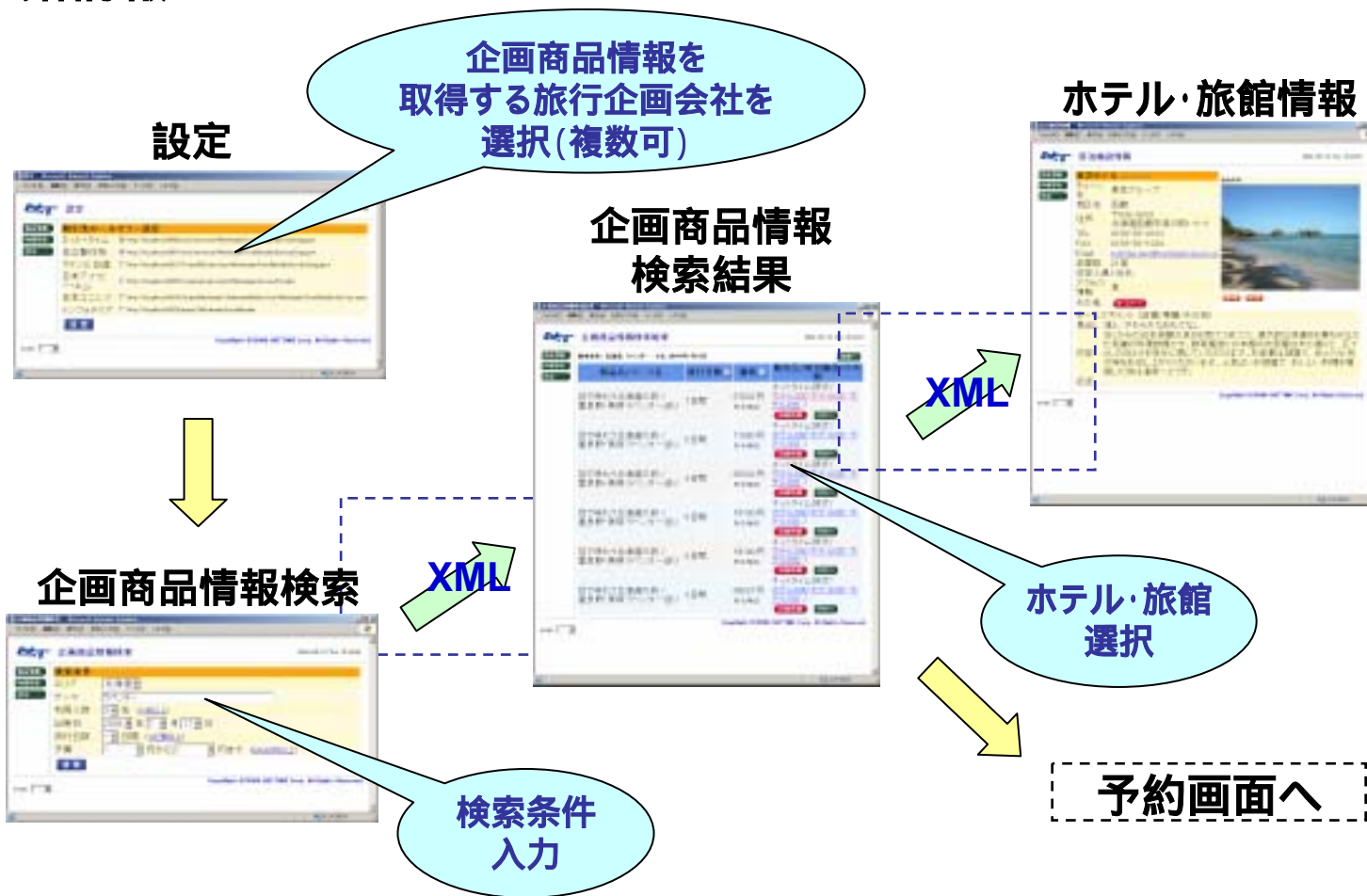
# システム構成

Operating System	Windows XP Professional
Development Language	Java 2 sdk 1.4.2_04
Web Application Server	Tomcat 4.1.30
Web Application (GUI)	JSP1.2, Servlet2.3
Web Service	Apache AXIS 1.1



# 画面フロー (1 / 4)

設定 企画商品情報検索 企画商品情報検索結果 ホテル・旅館情報



# 画面フロー (2 / 4)

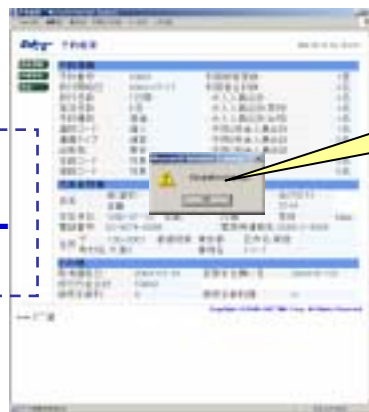
## 予約 予約結果

予約

予約結果



TravelXML



予約結果  
メッセージを表示

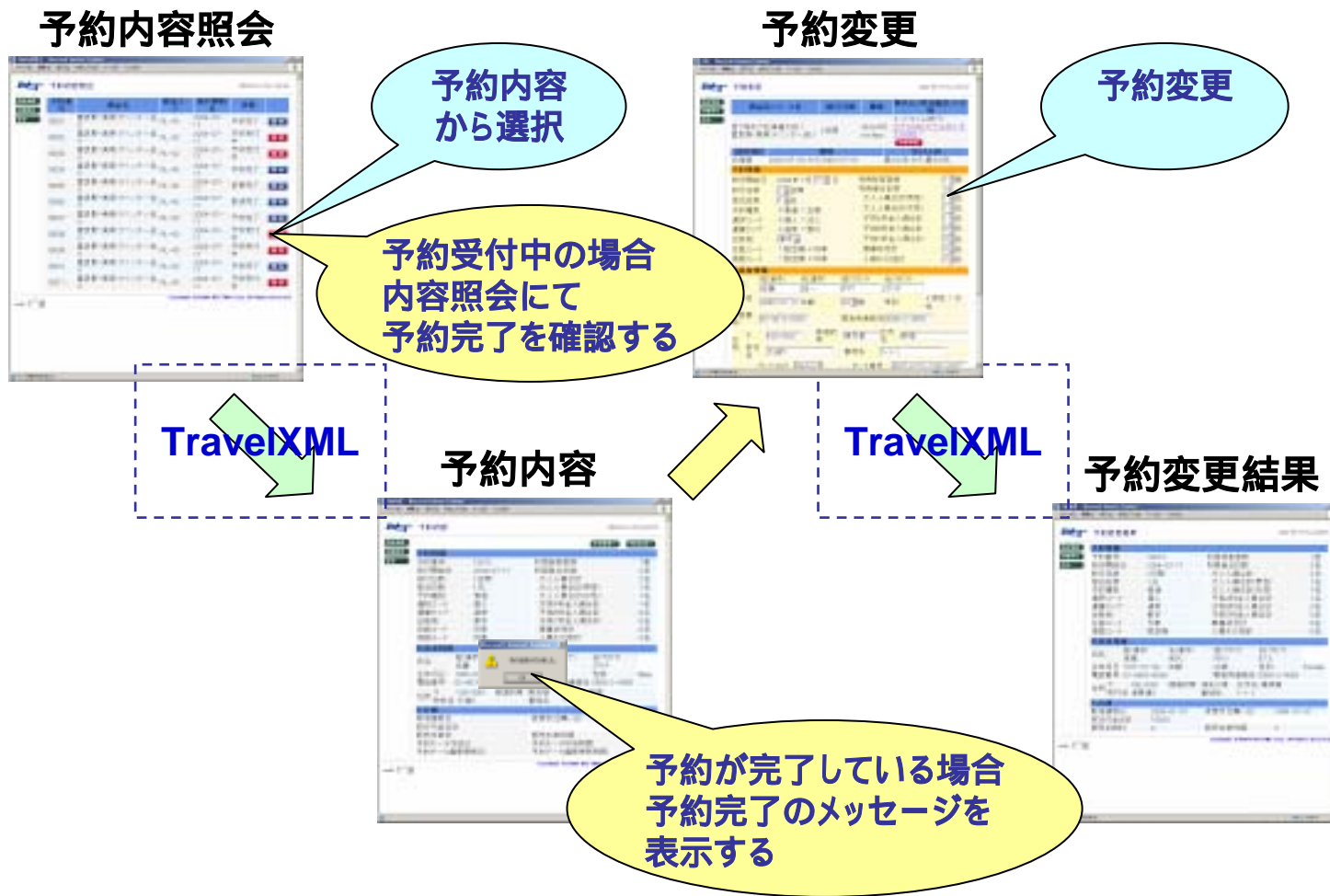
予約内容照会へ

予約情報  
入力

セキュリティ情報  
の署名・暗号化

# 画面フロー ( 3 / 4 )

予約内容照会    予約内容    予約変更    予約変更結果



XML Consortium

# 画面フロー ( 4 / 4 )

予約内容照会    予約内容    予約取消    予約取消結果





# 実装のポイント(1 / 2)

- 画面入力
  - 画面で入力された値をServletで取得し送信メッセージを作成する

例えば、予約画面で名前を入力する

氏名	姓(漢字)	名(漢字)	姓(フリガナ)	名(フリガナ)
	鈴木	良夫	スズキ	ヨシオ

HTMLソース

```
<input type="text"
  name="BookingRequest/BookingRequest Information/SurNameKanji"
  value="鈴木"/>
<input type="text"
  name="BookingRequest/BookingRequest Information/GivenNameKanji"
  value="良夫"/>
```

送信メッセージ

```
<BookingRequest>
  <BookingRequest Information>
    <SurNameKanji>鈴木<SurNameKanji>
    <GivenNameKanji>良夫<GivenNameKanji>
  <BookingRequest Information/>
</BookingRequest/>
```

ServletでPOSTされたパラメータ名(要素指定)から送信メッセージを作成する。クラスを判断して値を設定する。すべてのPOSTデータを判断して送信メッセージを作成する！

# 実装のポイント(2 / 2)

- 画面出力
  - 受信メッセージから要素を指定して値を取得し文字列として表示する

例えば、予約結果画面で旅行開始日の値を出力する

受信メッセージ(JSPでBeanとして取得する)

```
<BookingRequestReply>
  <BookingRequestInformation>
    <DepartureDate>2004-07-17<DepartureDate/>
  <BookingRequestInformation/>
</BookingRequestReply/>
```

旅行開始日はDateクラスとして受け取る

JSPソース

```
<%=Msg.getString(receiveMsg,
  "BookingRequestReply/BookingRequestInformation
  /DepartureDate)"%>
```

受信メッセージから旅行開始日の要素を指定する。クラスを判断して値を文字列として取得する。すべての受信データに対して文字列として出力する！

予約番号	10001		
旅行開始日	2004-07-17	利用者合計数	3名
旅行日数	1日間	大人人員合計	2名

# まとめ

## ■ 苦労した点

- 取り扱うデータ量が多い
  - 画面入出力で工夫が必要
  - データの解釈を明確にする
- セキュリティの実装
  - JDKのバージョンにより動かない場合があった
- 接続実験
  - 接続実験で気づく問題点が多かった(ライブラリによるスキーマタイプの違いなど)
  - 共有できるテスト環境の構築が早い時期に必要

## ■ 感想、その他

- 実ビジネスへの適用
  - End-to-EndのセキュリティによりWebサービス適用ビジネス拡大へ
  - SOAPハンドラの利用でセキュリティ処理を分離して実装
  - 証明書を利用したセキュアなビジネス展開に期待(ICカードソリューションとの結合)
- 社内へのフィードバック
  - 活動内容を実ビジネスへどのように適用するか
- XMLコンソーシアムの活動
  - 他企業間の交流がとても勉強になった
  - 技術的な知識の蓄積に役に立った



XML Consortium

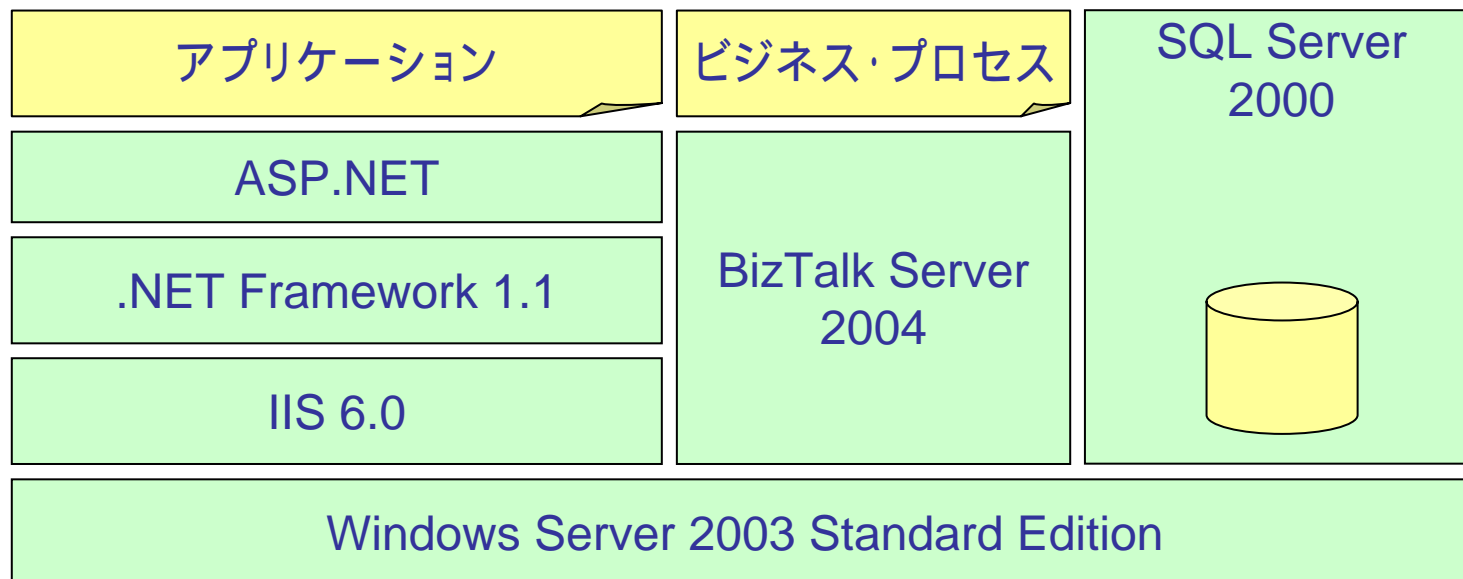
A decorative graphic consisting of a vertical black line intersecting a horizontal black line. To the left of the intersection, there are overlapping colored squares: a purple square on top and a green square on the bottom, both with a gradient effect.

## 旅行企画会社システム

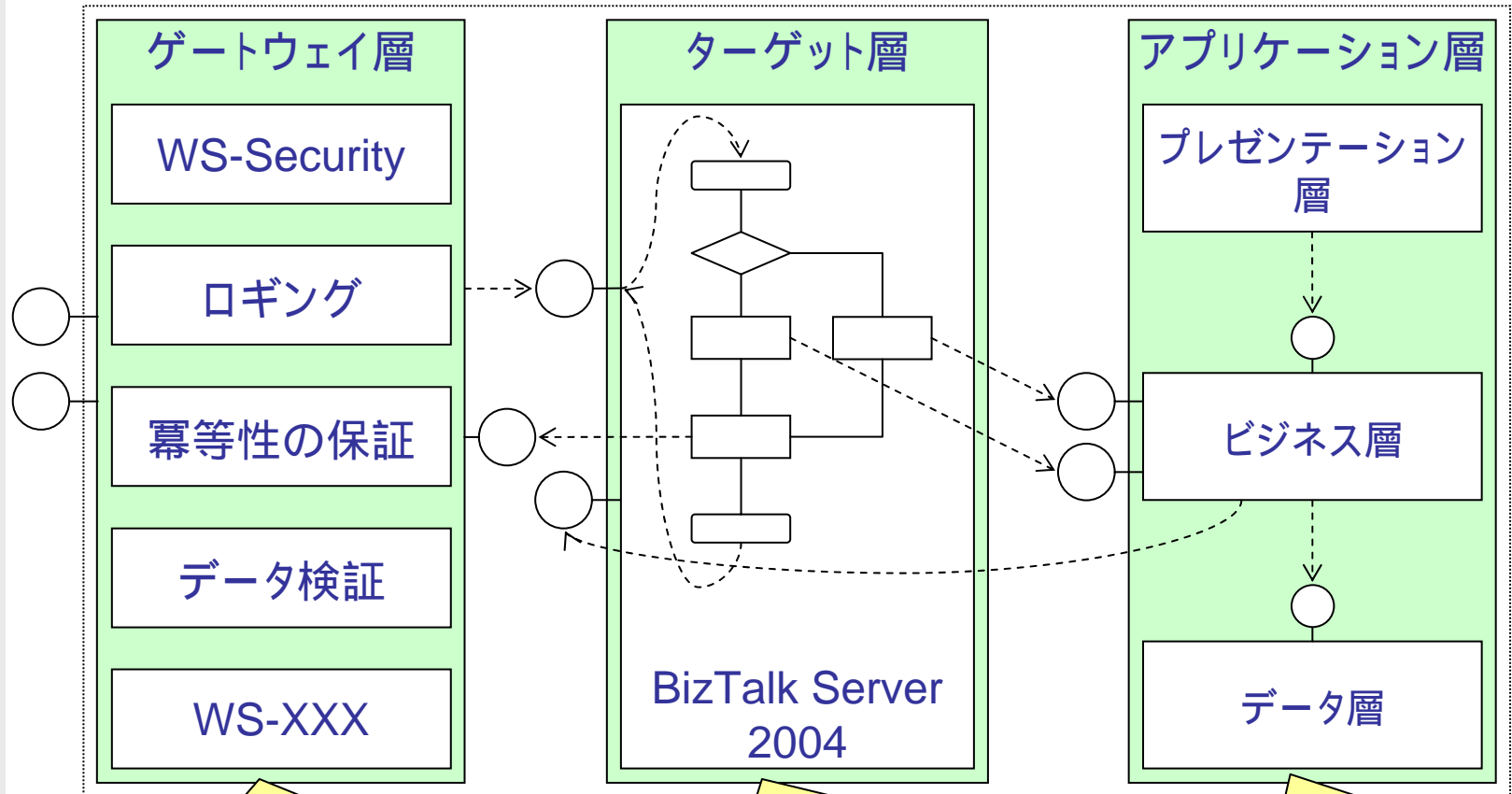
日本ユニシス株式会社 尾島 良司

- 日本ユニシスの旅行企画会社システムの特徴。
  - 三層に分割された企業システム・アーキテクチャ。
  - Microsoft BizTalk Server 2004。
  - Microsoft .NET Framework 1.1。

## ■ システム構成



- 三層に分割された企業システム・アーキテクチャ



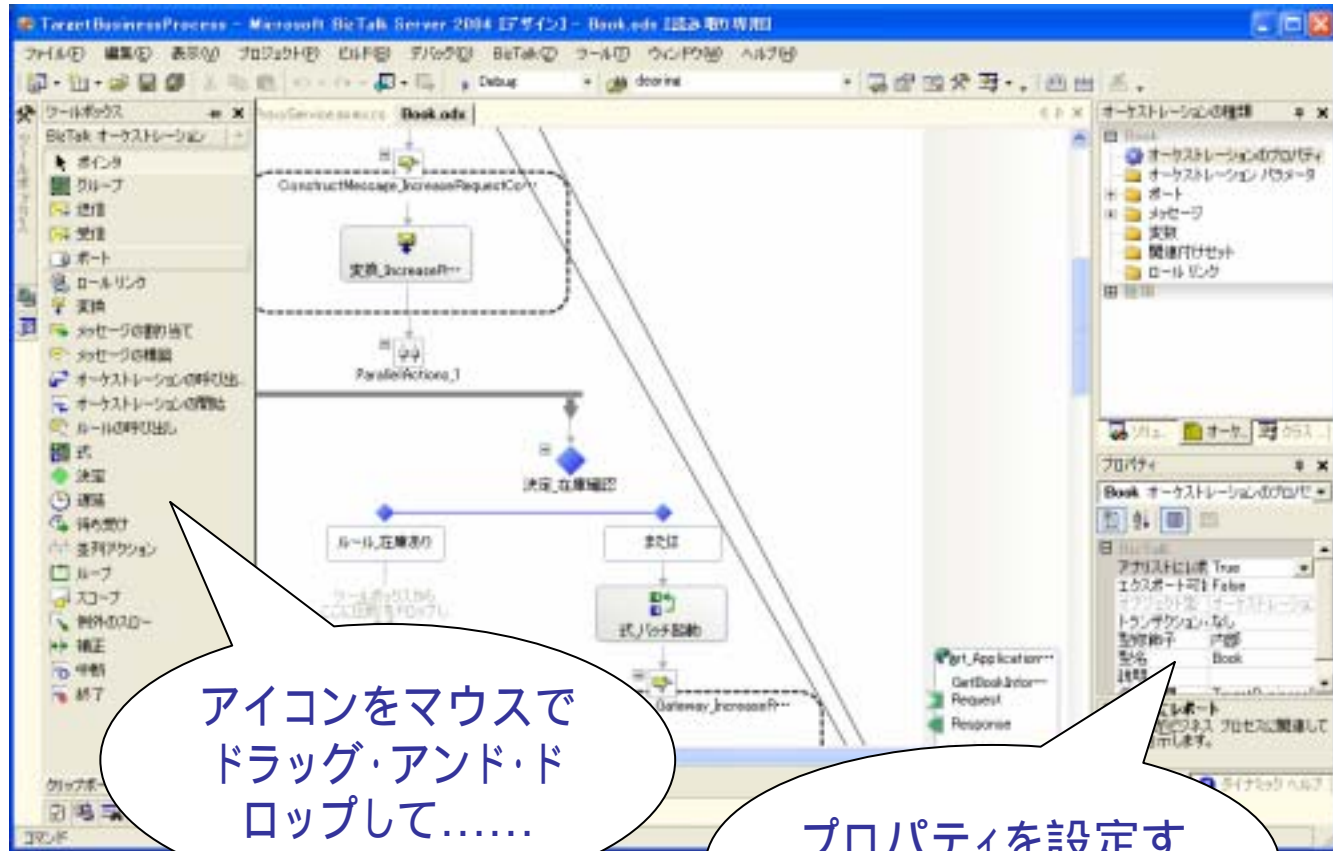
最新のWebサービスに  
対応

ビジネス・プロセス管理  
プライベート・メッセージに変換

EAのService Component  
ここも三層で作成する

- Microsoft BizTalk Server 2004
  - EAI (Enterprise Application Integration) ソフトウェア。ビジネス・プロセスを定義し、実行・管理する。
  - 並行処理、分岐、繰り返しなどの実行制御と、メッセージの変換をビジネス・プロセスに組み込める。
  - 通信相手が停止していても、実行途中にコンピュータをリブートしても、ビジネス・プロセスを確実に実行、メッセージを確実に配送する。
  - ロング・トランザクションをサポートする。
  - XML、Webサービス、そしてBPEL4WS (Business Process Execution Language for Web Services) をサポートする。
  - BAM (Business Activity Monitoring)。ビジネス・イベントをキャッチし、レポート。経営判断に役立てる。
  - そして、プログラミングが非常に簡単。ドラッグ・アンド・ドロップするだけ。

- Microsoft BizTalk Server 2004でのプログラミング



アイコンをマウスで  
ドラッグ・アンド・ド  
ロップして.....

プロパティを設定す  
るだけ。



- Microsoft .NET Framework 1.1
  - 実装効率が良い。
    - 今回のデモの実装は1週間 × 1人 (BizTalk Serverでの実装を含む)。
    - GUIの実装は30分。
  - Webサービスにネイティブに対応。
    - IISの上で動作。
    - ライブラリの追加は不要。
  - ただし、一部に納得がいけない仕様も。
    - .NET FrameworkのWebサービスのデフォルトはSOAPActionでバインディング。SOAPメッセージでバインディングさせるにはカスタム属性を指定しなければならない。
    - 一行書けばJavaとつながるので、実害はないけど。それにしてもMicrosoftはWS-Iの中核企業なのに.....。
  - 今回の実証実験で.NET Frameworkを採用したのは日本ユニシスだけで、とても寂しかったです。
  - Webサービスを使えば相互運用でき、三層モデルで共存可能になるので、それぞれの特性に合わせて使い分けてゆきませんか？

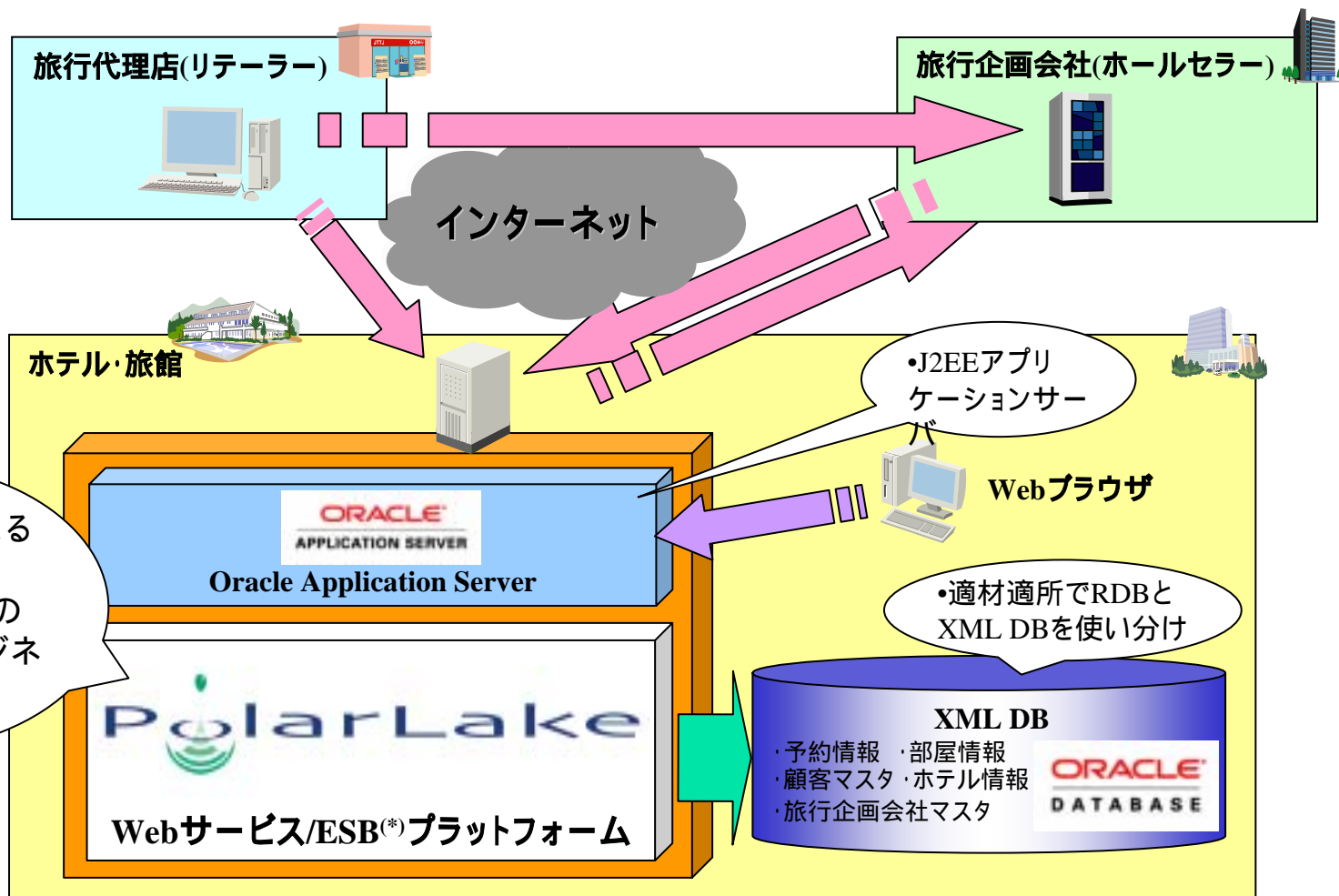
A decorative graphic on the left side of the slide, consisting of a vertical black line and a horizontal black line intersecting. The top-left quadrant is a purple square, and the bottom-left quadrant is a green square, both with a gradient effect.

## ホテル・旅館システム

株式会社日立システムアンドサービス 赤澤 一憲  
日本オラクル株式会社

# 株式会社日立システムアンドサービス 日本オラクル株式会社

XML Consortium



•サービス指向によるB2B/EAIを実現  
•コンテンツベースのルーティングやビジネスプロセス制御

•ESB: エンタープライズ・サービス・バス(Enterprise Service Bus)

•PolarLakeは、アイルランドPolarLake社の商標です。  
•Oracle、Oracle Database 及び Oracle Application Serverは、  
米国Oracle Corporationの登録商標 又は商標です。

# システム構成

## ■ 使用製品

OS	Windows 2000 Professional
Webサービス	PolarLake Database Integrator with Mapper Module 3.2.6
XML DB	Orcle9i Database (9.2.0.1.0)
APサーバ	Oracle Application Server 10g (9.0.4)
開発言語	Java 2 SDK 1.4.2

## ■ 開発規模

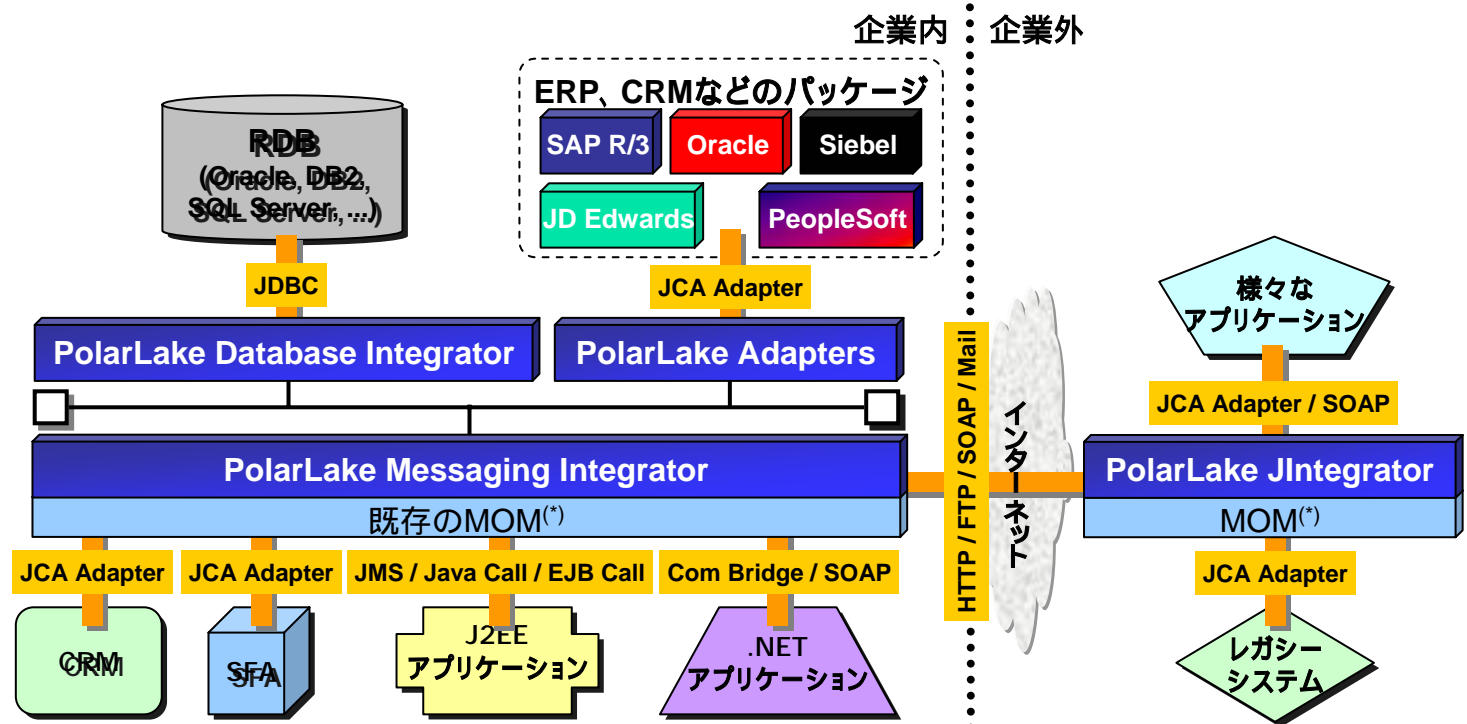
開発規模	4.0 キロステップ
使用工数	20人日

\* 大部分が、Oracle XML DB対応部分

# 株式会社日立システムアンドサービス 日本オラクル株式会社



- PolarLakeは、新世代のB2B/EAIの手法として注目されているエンタープライズ・サービス・バス(ESB\*)の要件に最も合致した製品
  - エンタープライズ・サービス・バス(ESB)とは・・・  
標準規格の技術を利用した、サービス指向アーキテクチャによるアプリケーション間の連携や統合を実現するための、新しい製品分野・手法
  - そのメリットは・・・  
低コストかつ低リスクで、システム連携・統合を実現できる



\*ESB: Enterprise Service Bus  
\*MOM: Messaging Oriented Middleware

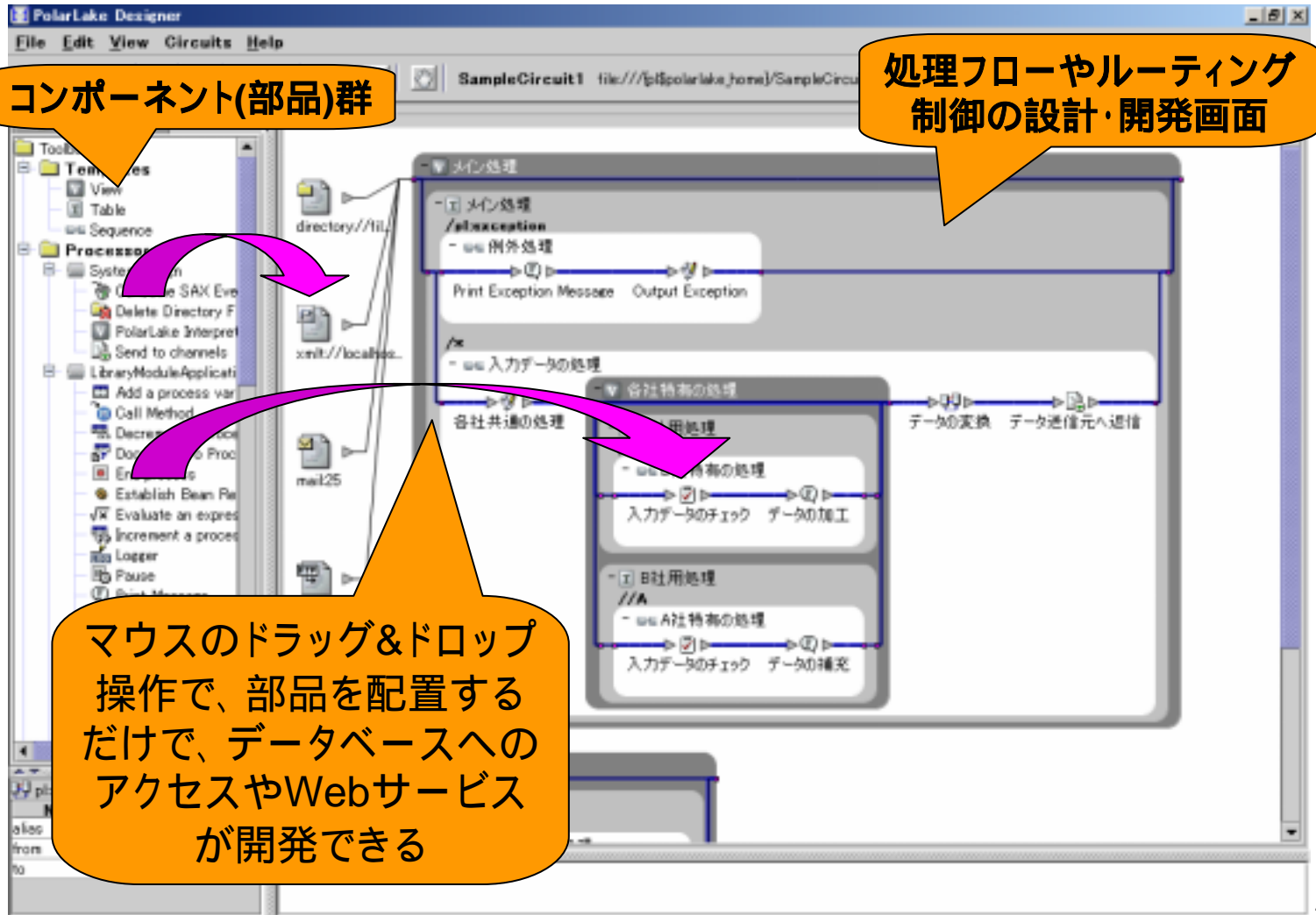
(1)

- 処理フローやルーティングの設計・開発

コンポーネント(部品)群

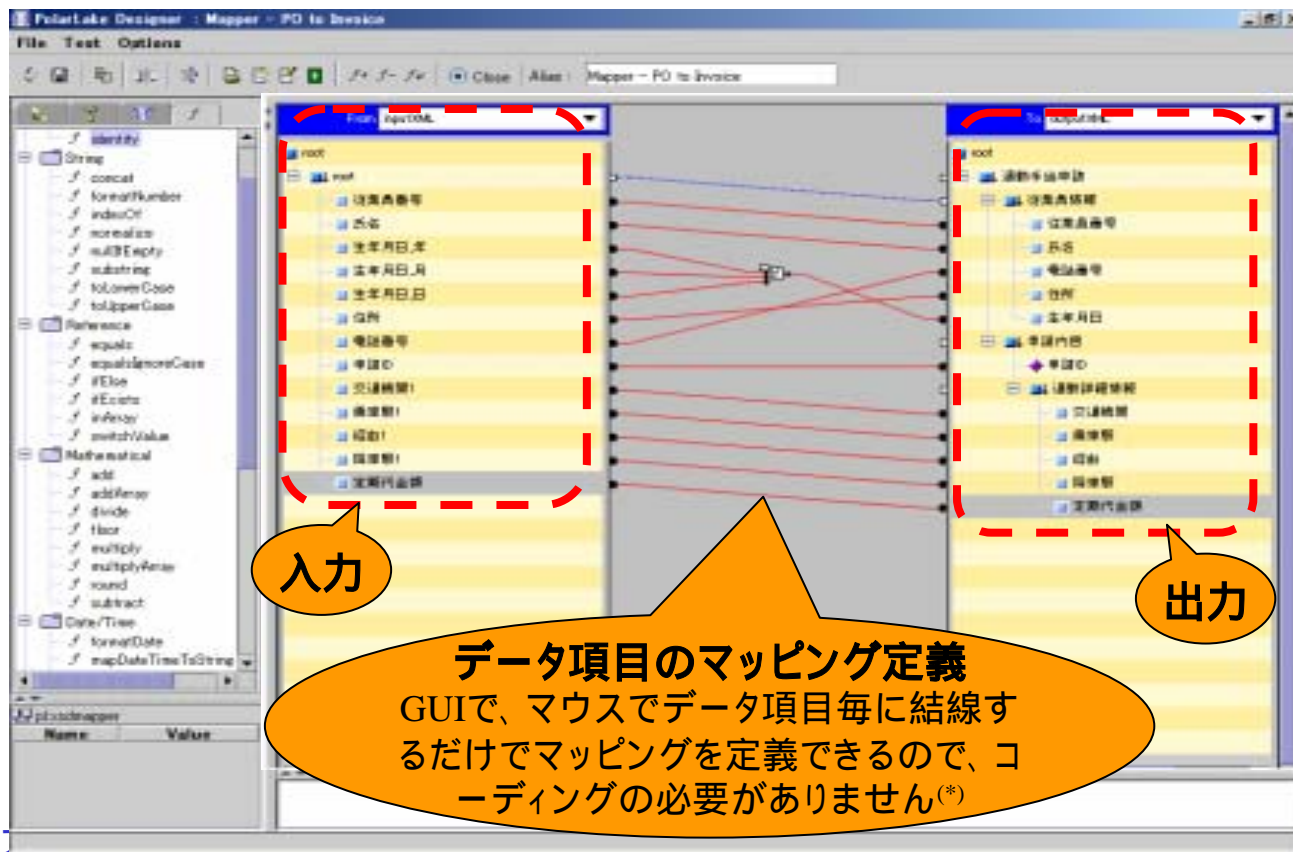
処理フローやルーティング  
制御の設計・開発画面

マウスのドラッグ&ドロップ  
操作で、部品を配置する  
だけで、データベースへの  
アクセスやWebサービス  
が開発できる



(2)

- マッピング・モジュールでの各種データのマッピング定義
  - 処理フロー上に流れるXML XMLデータ形式変換
  - 処理フロー上に流れるXMLの要素 RDBのカラムのマッピング
  - 処理フロー上に流れるXMLの要素 SOAPリクエスト/レスポンスの要素のマッピング



# 成果・感想等

- 苦労した点
  - データ項目が多かったり、よく似た項目も多かったため、個々のデータ項目について、理解が難しかった。また、他社との連携に際し、個々のデータ項目の詳細な仕様について、認識のズレが発生したため、すぐに接続できないこともあった。
  - PolarLakeとApache AxisでNamespaceの扱い方が違っていたために、最初、接続できないことがあり苦労した(両方に正しくNamespaceを指定することで連携できた)。
  - 今回、初めてOracle XML DBを使用した。利用方法の検討に苦労した(RDBとの使い分けの点で、若干、ノウハウ不足の点があった)
- 成果
  - PolarLakeがWebサービスで多くの他社製品と接続出来る事を確認できた。
  - WS-Securityの利用、実装のノウハウを得ることができた。
  - Oracle XML DBの利用ノウハウを得ることができた。
- 感想
  - 今回、他社との接続が初めてだったり、初めてOracle XML DBを使用したため、苦労も大きかったが、ノウハウとしてたまったので、今後に活かしていきたい。
  - 今後もこのような、接続実験に参加していきたい。
  - TravelXMLのデータ項目が、もっと整理されたものにならないか。





XML Consortium

つづく