

【TravelXML利用Webサービス実証実験プロジェクト成果資料】

A decorative graphic consisting of a vertical black line and a horizontal black line intersecting at the origin. The top-left quadrant is filled with a purple-to-white gradient, and the bottom-left quadrant is filled with a green-to-white gradient.

# WSDL策定について

応用技術部会 WebサービスWG  
アドソル日進株式会社 荒本 道隆

# WSDL策定のポイント

- スキーマは、すでにTravelXMLにて策定済み
  - TravelXML1.1.1勧告をそのまま import して使用
  - さらに今回のデモで必要なスキーマを追加
  - スキーマが変更になっても、operationが増えない限り、WSDLを修正する必要はない
- Document/Literalを採用
- APIとServiceを分離して定義
  
- 作成方法
  - テキストエディタで作成し、WSDL2Java (AXIS) やwsdl.exe (.NET) で検証
    - 特定の実装に依存しないように

# WSDL策定時の課題-1

- 名前空間の割り当て
  - それぞれに異なる名前空間を割り当てた
    - TravelXML1.1.1  
<http://www.xmlconsortium.org/bukai/ouyou/demo/travel>
    - デモ用スキーマ  
<http://www.xmlconsortium.org/bukai/ouyou/demo/PackTour>
    - API定義  
<http://www.xmlconsortium.org/wg-app-web/TravelXML/TravelAPI>
    - ホールセラー用サービス  
<http://www.xmlconsortium.org/wg-app-web/TravelXML/WholesalerService>
    - 宿泊施設用サービス  
<http://www.xmlconsortium.org/wg-app-web/TravelXML/AccommodationService>
  - 特に、異なるサービスに同じ名前空間を割り当てると、コードの自動生成時に同じ場所に上書きされてしまう
- 1つの名前空間の定義は、1つのファイルで
  - WSDL.exe(.NET)は、同じ名前空間の物をimportすると問題アリ

## WSDL策定時の課題-2

- element名と同じoperation名を使う場合
  - element名の先頭が大文字なので、operation名の先頭を小文字に
  - まったく同じだと、WSDL2Java(Axis)で問題アリ
- binding
  - 通信相手により、分類
  - メッセージの種類が増えても、クラスは増えない、メソッドが増える
- service
  - 実装するサーバごとに、分類

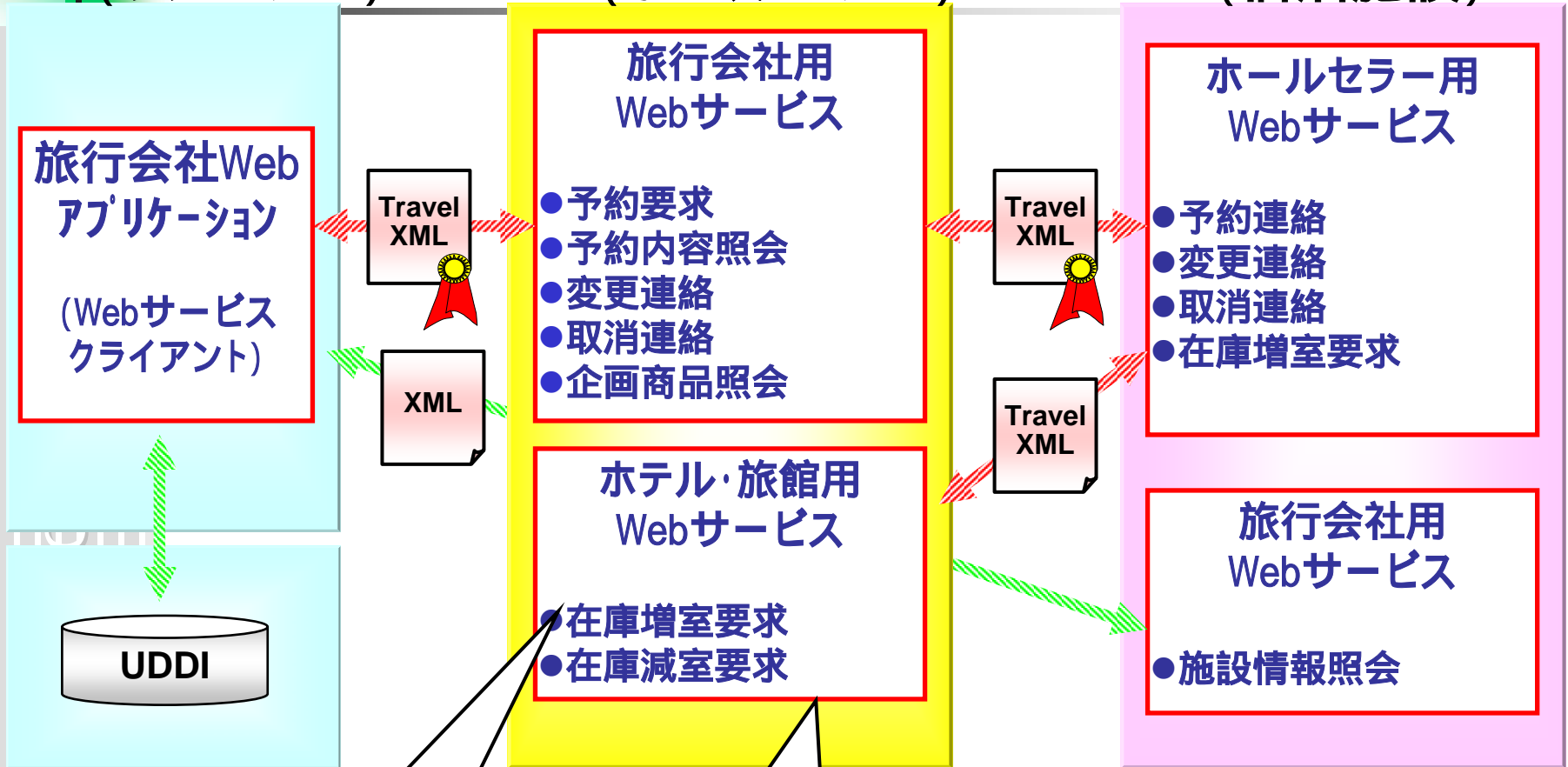
# Webサービスの配置構成



旅行会社  
(リテラー)

旅行企画会社  
(ホールセラー)

ホテル・旅館  
(宿泊施設)



operation単位

binding単位

service単位

# 成果・感想等

- スキーマとWSDLを完全に分離
  - 既存のスキーマをそのまま使う事で、省力化
  - スキーマの変更があっても、WSDLは変更しなくても済む
- 名前空間
  - どの単位で名前空間を分割するか？
  - 各実装で、動きがかなり違う

いつも名前空間の扱いには苦労する....