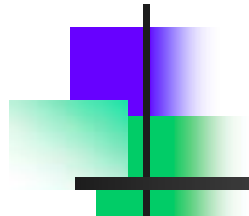


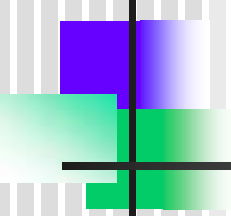
システム概要とシステムモデル説明

A decorative graphic on the left side of the slide, consisting of a black crosshair with a purple square in the top-left quadrant and a green square in the bottom-left quadrant.

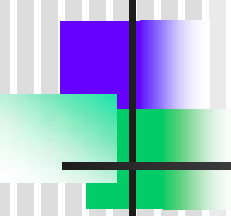
2003年5月29日

富士通 松本 賢一

日本ユニシス 牧野 友紀

A decorative graphic consisting of a black crosshair overlaid on a blue and green gradient square.

1. システム概要

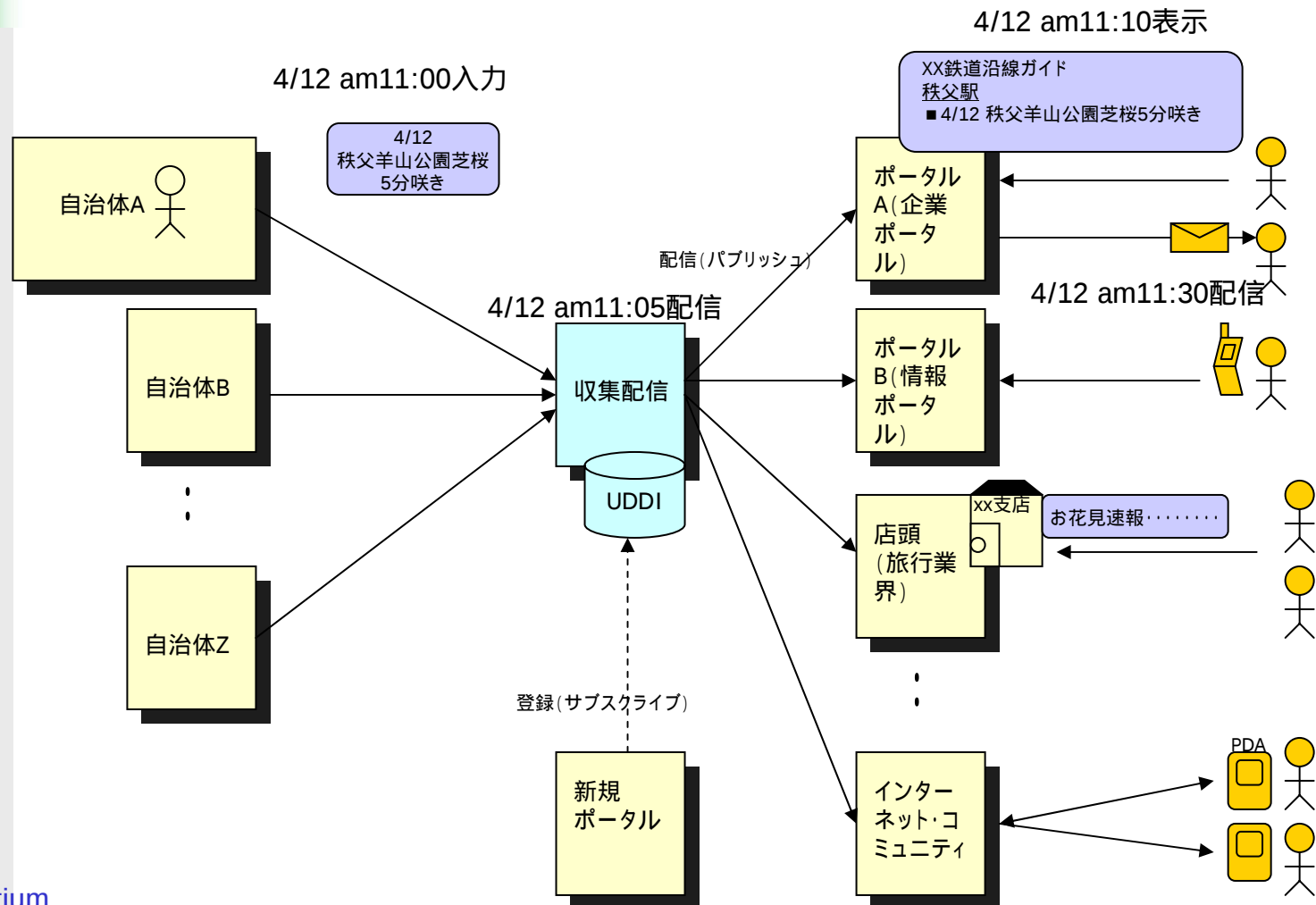


1) システム特徴

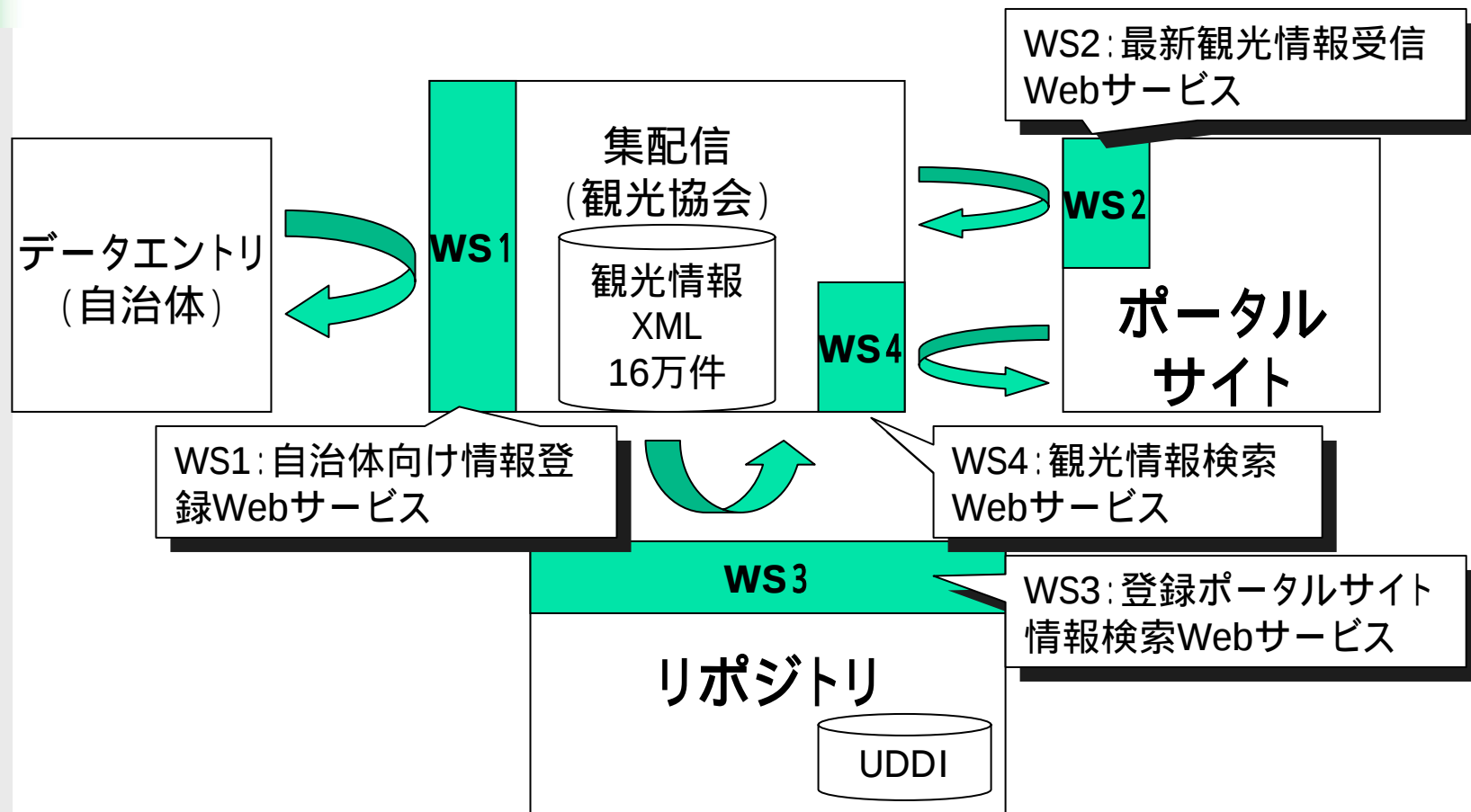
企業間を跨る情報の収集・配信をWebサービスで実装

- 情報発信元からポータルサイトへ,ポータルサイトから利用者へリアルタイムに配信される。
- 一回の情報配信で,不特定多数の利用者に提供される。
- 同一の情報が,複数の異なる目的の情報サイトで利用される。
- 同一の情報が,複数の異なる形式の端末で利用される。
- ポータルサイトの参加登録はパブリッシュ/サブスクライブ・モデルの自動オンライン登録

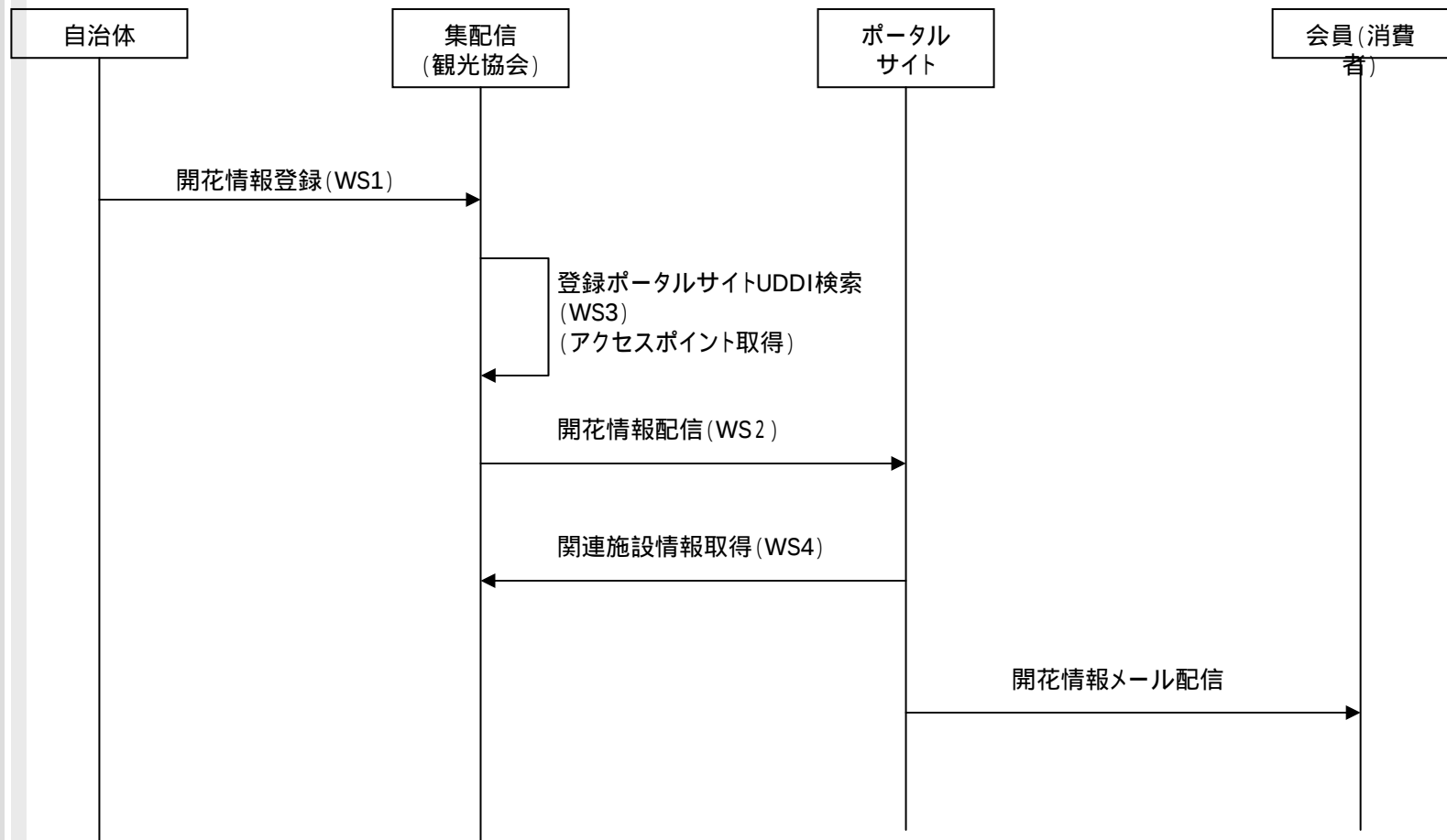
2) 情報集配信概念図



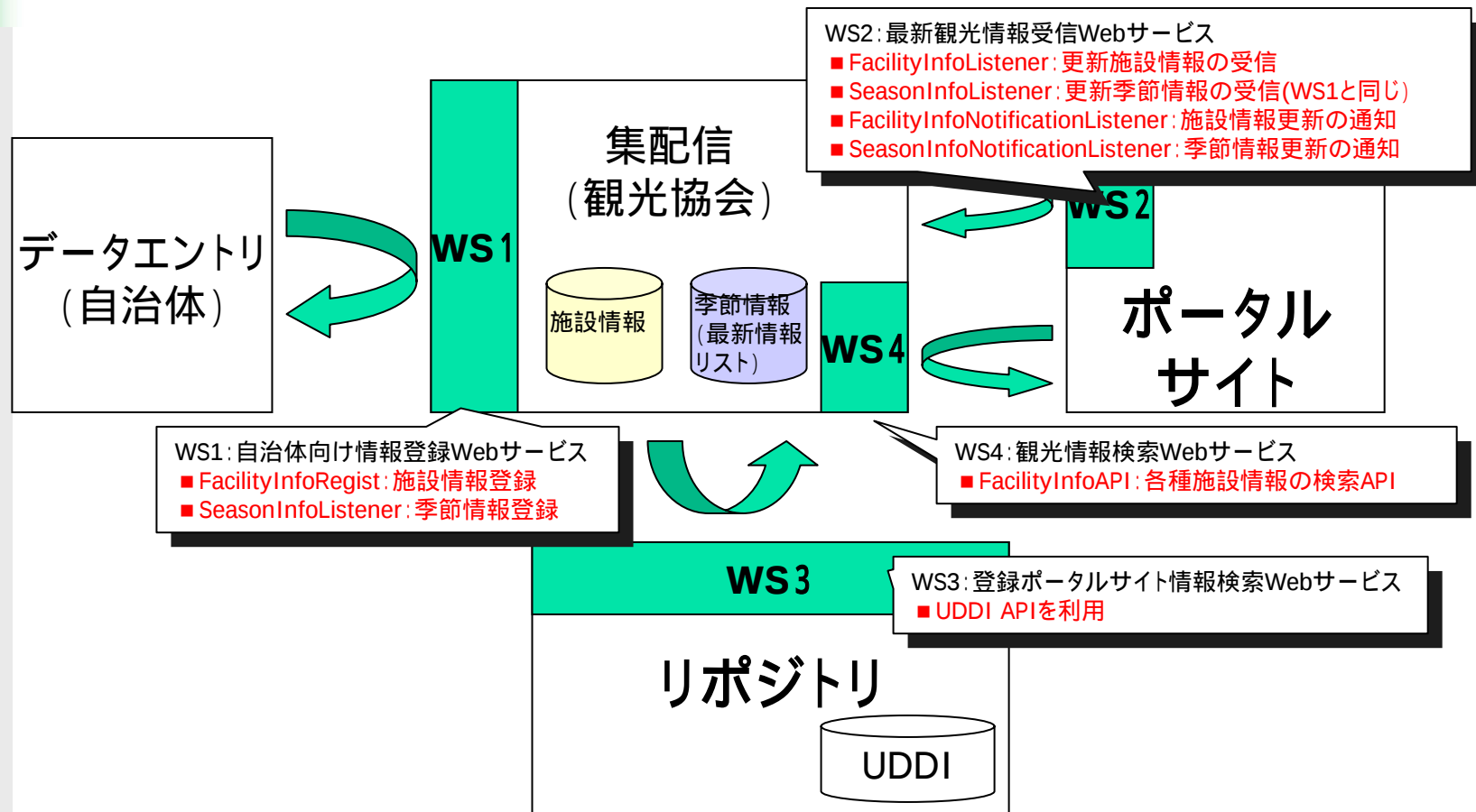
3) システム・モデル



4) 情報の流れ例 (季節情報)



5) Webサービス配置



6) Webサービス一覧 (WS1, WS2)

- WS1 (自治体向けサービス)
 - FacilityInfoRegist:
施設情報を登録する。一つの施設情報を登録することが可能。
 - SeasonInfoListener:
季節情報を更新する。同時に複数の季節情報をアップロードすることが可能。
- WS2 (ポータル側実装)
ポータルサイトでは、以下の4つを選択して実装する
 - FacilityInfoListener:
このWebサービスを公開し、UDDIに登録することで、施設情報が更新されたとき、施設抜粋一覧を受信することが可能。
 - SeasonInfoListener:
このWebサービスを公開し、UDDIに登録することで、季節情報が更新されたとき、更新分の季節情報一覧を受信することが可能。
 - FacilityInfoNotificationListener:
このWebサービスを公開し、UDDIに登録することで、施設情報が更新されたとき、通知(データなし)を受けることが可能。
 - SeasonInfoNotificationListener:
このWebサービスを公開し、UDDIに登録することで、季節情報が更新されたとき、通知(データなし)を受けることが可能。

7) Webサービス一覧 (WS4)

■ WS4 (ポータル向けサービス)

FacilityInfoAPI:

ポータルサイト(など)が、観光情報を検索したり、取得したりするために使います。次の6つの操作 (operation) が用意されている。

- FindItemsByArea:
関東、東京、といったエリアで絞りこんだ施設情報を検索する操作。結果として、施設抜粋一覧 (ItemList) を返す。
- FindItemsByGenre:
レジャー、スポーツ、観光などといったカテゴリーで施設情報を検索する操作。結果として、施設抜粋一覧 (ItemList) を返す。
- FindItemsByMeshGI:
メッシュ値 (位置情報) から施設情報を検索する操作。結果として、施設抜粋一覧 (ItemList) を返す。
- FindItemsByTime:
ある特定の時刻より新しい施設情報を検索する操作。結果として、施設抜粋一覧 (ItemList) を返す。
- GetItemInfo:
施設抜粋一覧 (ItemList) を渡すと、そこに入っている 施設情報 (Facility) を一覧して返す。
- GetSeasonInfo:
施設抜粋一覧 (ItemList) を渡すと、そこに入っている 施設に関連した季節情報を一覧して返す。

8) 技術面での特徴

- Document/Literal型SOAPバインディングの利用
 - 観光情報XML (FacilityInfo.xsd) など既存のXMLドキュメントをSOAPにより転送
 - XMLスキーマ (xsd) に基づくデータ・エンコーディング
- UDDIを用いたパブリッシュ/サブスクライブ型情報配信モデルの実現
 - ポータル・サイト情報をプライベートUDDIで管理。
UDDI API等によりオンラインで登録後, 直ぐに配信開始
 - インタフェース型をtModelで管理し, 一意性・識別性を持つ
 - ポータルサイトの希望する配信情報種選択指定にタクソノミ-を利用。タクソノミ-によりポータルサイト一覧を逆引き
- ワンソース・マルチデバイス
 - XMLで表現された同一情報を, Webブラウザ, 携帯電話, PDA, 電子メール, 音声にて視聴。

9) WSDL策定ポリシー

WSDL 3層構造の各層分離

- スキーマ:types
 - 交換するXMLドキュメントの定義。特定のWebサービスに依存しない普遍的な情報や, APIで入出力するメッセージを構成するXMLフラグメントの定義。
 - 施設情報(観光協会XML): FacilityInfo.xsd
 - 季節情報: SeasonInfoListener.xsd
 - 観光情報WebサービスAPI用XMLスキーマ: TourInfoAPI.xsd
- 論理API:message,portType,(binding)
 - サービス名, 入出力メッセージ, 構成データ型(スキーマを包含)などの論理的な定義
 - 複数の実装によらず一意。tModelで識別する単位。
 - 観光協会WebサービスAPI: TourInfoAPI.wsdl
- 実装API:binding,service
 - 各ポータルサイト, 各集配信で実装するWebサービスの記述
 - 日立ソフト社実装Webサービス: TourInfoAPIServiceXSC1.wsdl



11) UDDI利用モデル

本モデル

サービス提供者
(集配信)

提供

サービス利用者
(ポータル)

検索

UDDI

登録

利用形態が逆

一般的には...

サービス提供者

利用

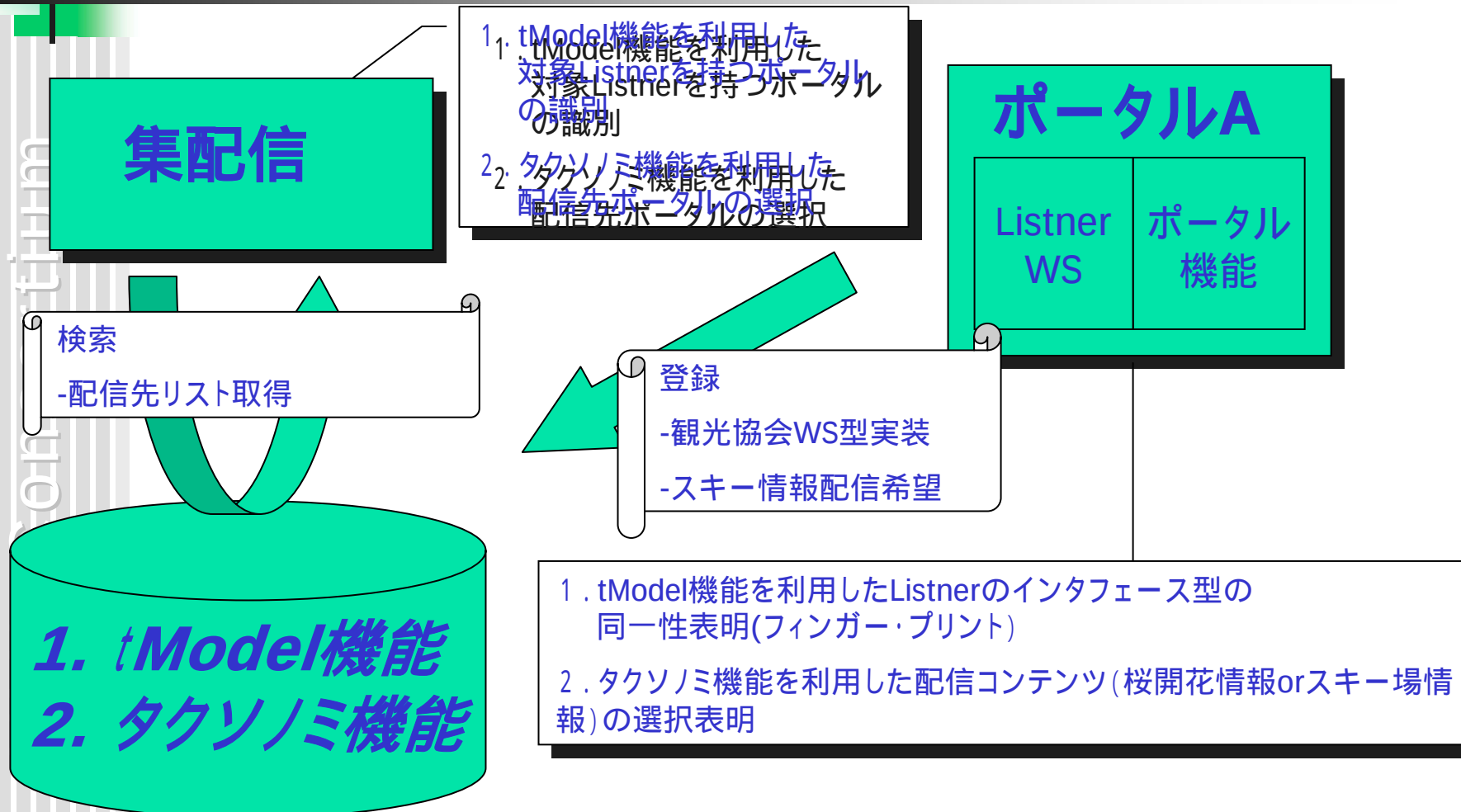
サービス利用者

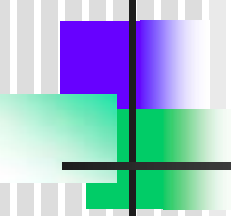
登録(公開)

UDDI

検索

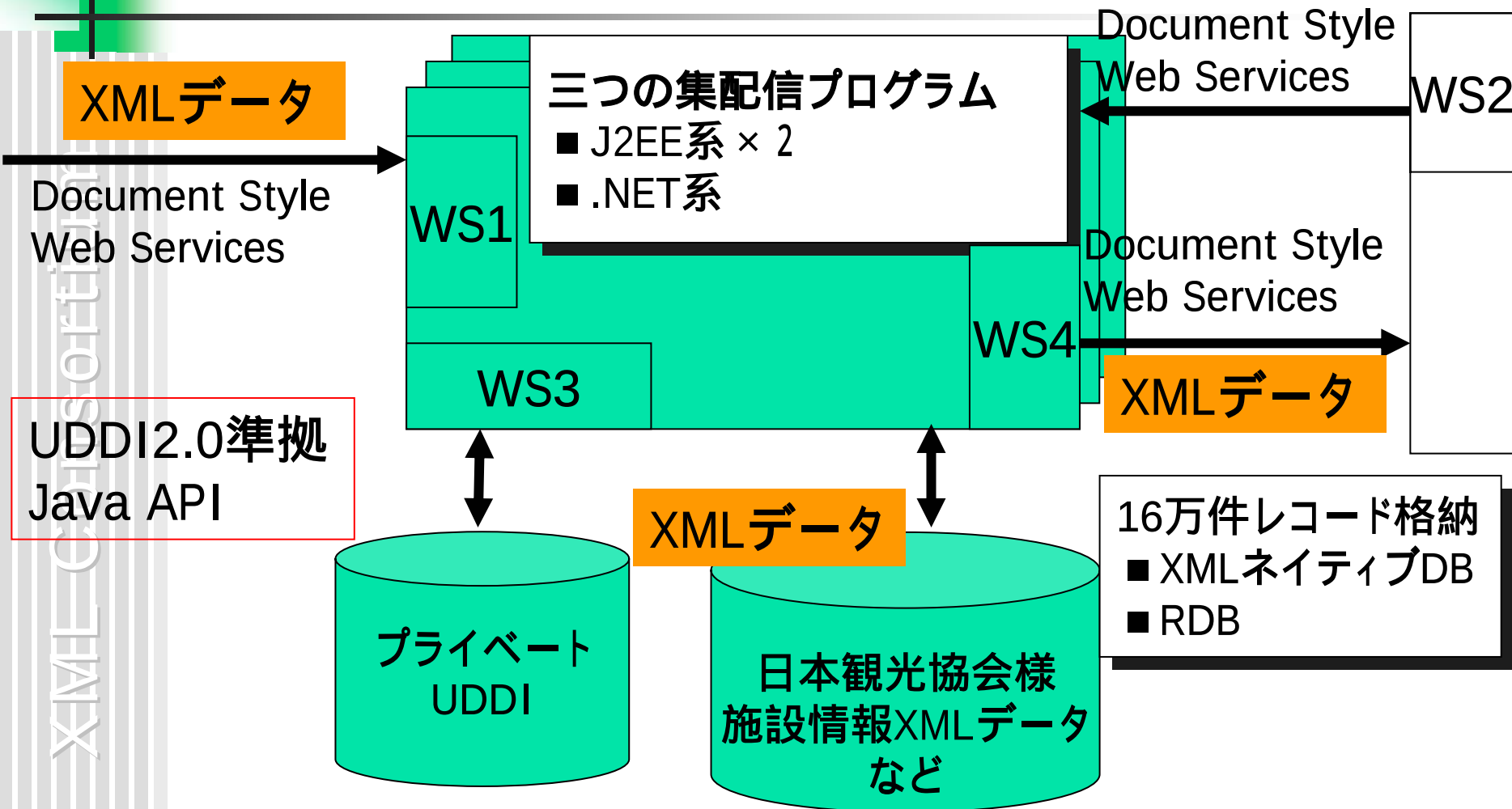
12) 利用するUDDIの機能



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

2 . 集配信システム

1) 集配信システム概要



2) 3つの集配信による相互接続性実験

- 集配信のすべての機能を3つのシステムに実装
 - J2EE系集配信プログラム × 2
 - .NET系集配信プログラム
- 別々のシステムに搭載されているWebサービスを同一インターフェースで呼び出し可能
- データ共有により異なるWebサービスを呼び出しても同じ結果が得られる

3) document/literal型の 相互運用性実証

- 各アプリケーションサーバーとの通信は document/literal 型を用いて実現
- 全てのアプリケーションサーバーと接続できたことで、 document/literal型の相互運用性 (interoperability)の実証に成功
- rpc/encoded型に比べ、変更が容易
- XML DBとの親和性が高い

4) XML DBの利用

- 16万件全てのデータをNative XML DBに格納
- document/literal型のWebサービスと組み合わせることで、XMLの利点を保存するまで利用
(DOMツリーを保存、SQLは使わない！)
- 製品内蔵のXML SchemaによるXMLデータの自動チェック (ValidなXMLの検証)
- 開発者はXMLパーサー関連の余計なプログラム省くことができ、パフォーマンスの向上も同時に図られる
(RDBマッピングは不要)

5) プライベートUDDIを用いた初の 実証実験

- UDDIを用いて各配信先のWebサービスを登録
 - 標準API (UDDIv2 API)の利用
 - プラットホーム, プログラミング・モデル非依存
- 登録された情報をもとに送るデータを選別
 - タクソノミー機能の利用(各社businessServiceに付与)
 - tModelKey="UUID:70D80F34-4D7A-4BF0-A2BC-89D82C4B16DF"
keyName="ski" KeyValue="001"
 - tModelKey="UUID:70D80F34-4D7A-4BF0-A2BC-89D82C4B16DF"
keyName="sakura"keyValue="002"
- 各登録されたUDDI情報はUDDI API(Webサービス)もしくは簡易HTML画面によりダイナミックに変更可能

6) Discovery URL

```

- <businessEntity businessKey="BB9BBEEF-DF6D-41CD-8072-AB8AC2610BBE" operator="localhost:8888/uddi"
authorizedName="ABB591CE1AB84E4CB70FB5A3ECD37CC6" xmlns="urn:uddi-org:api_v2" generic="2.0">
- <discoveryURLs>
  <discoveryURL useType="businessEntity">http://localhost:8888/uddi/inquiryget?businessKey=BB9BBEEF-DF6D-41CD-8072-AB8AC2610BBE
  </discoveryURL>
</discoveryURLs>
<name>日本ユニシス株式会社</name>
<description>豊洲にあります</description>
- <contacts>
- <contact>
  <personName>小林茂</personName>
  <phone useType="voice">03-9999-9999</phone>
  <email>kobayashi@king</email>
</contact>
</contacts>
- <businessServices>
- <businessService serviceKey="6158CD9A-3CFE-4A55-8570-CC02D7A8A452" businessKey="BB9BBEEF-DF6D-41CD-8072-AB8AC2610BBE">
  <name>SeasonInfoListener</name>
  <description>スキー場の積雪情報をサービスします。</description>
- <bindingTemplates>
- <bindingTemplate serviceKey="6158CD9A-3CFE-4A55-8570-CC02D7A8A452" bindingKey="27A85D31-6FF7-4A00-B173-9DF76D586D4C">
  <description>スキー場積雪情報サービスをSOAPで提供</description>
  <accessPoint URLType="http">http://172.16.3.90/TravelAgencyWS2Test1/Service1.asmx</accessPoint>
- <tModelInstanceDetails>
  <tModelInstanceInfo tModelKey="UUID:5516BF35-23E2-493A-9F76-DB9231DA0D0B" />
  </tModelInstanceDetails>
</bindingTemplate>
</bindingTemplates>
- <categoryBag>
  <keyedReference tModelKey="UUID:70D80F34-4D7A-4BF0-A2BC-89D82C4B16DF" keyName="ski" keyValue="001" />
</categoryBag>
</businessService>
</businessServices>
</businessEntity>

```

White Page

Green Page

Yellow page

UDDI Inquiry Point

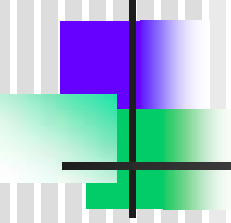


登録されている各社の情報



登録されているサービスの詳細情報

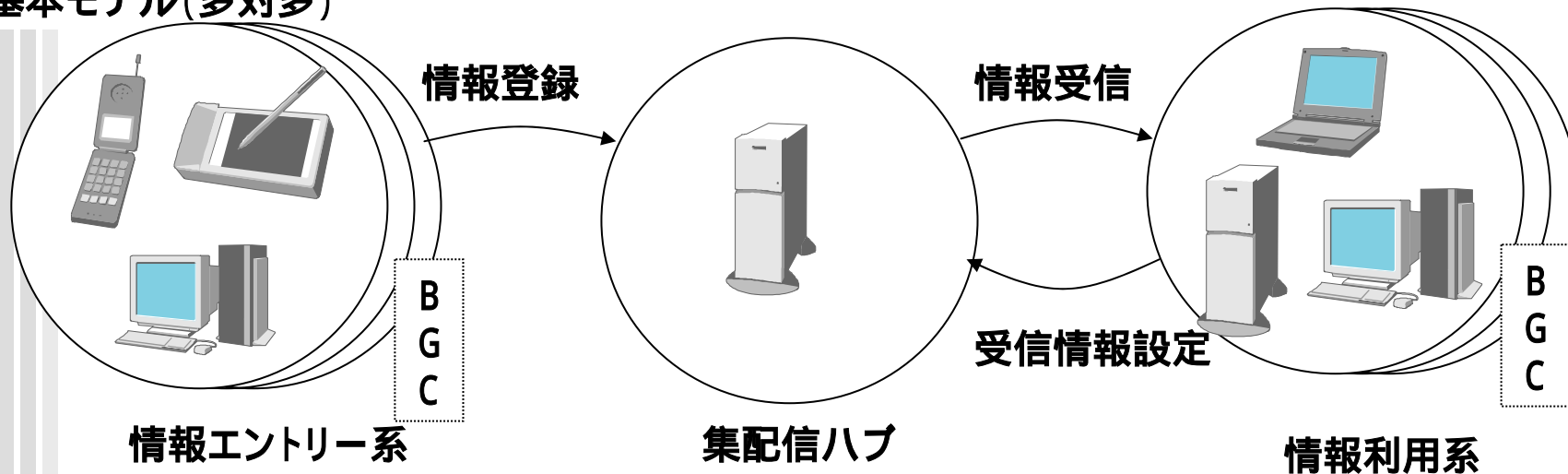


A decorative graphic consisting of a black crosshair overlaid on a blue and green gradient square.

3 . システムモデルの評価

1) 実証実験モデルの想定(特長)

基本モデル(多対多)

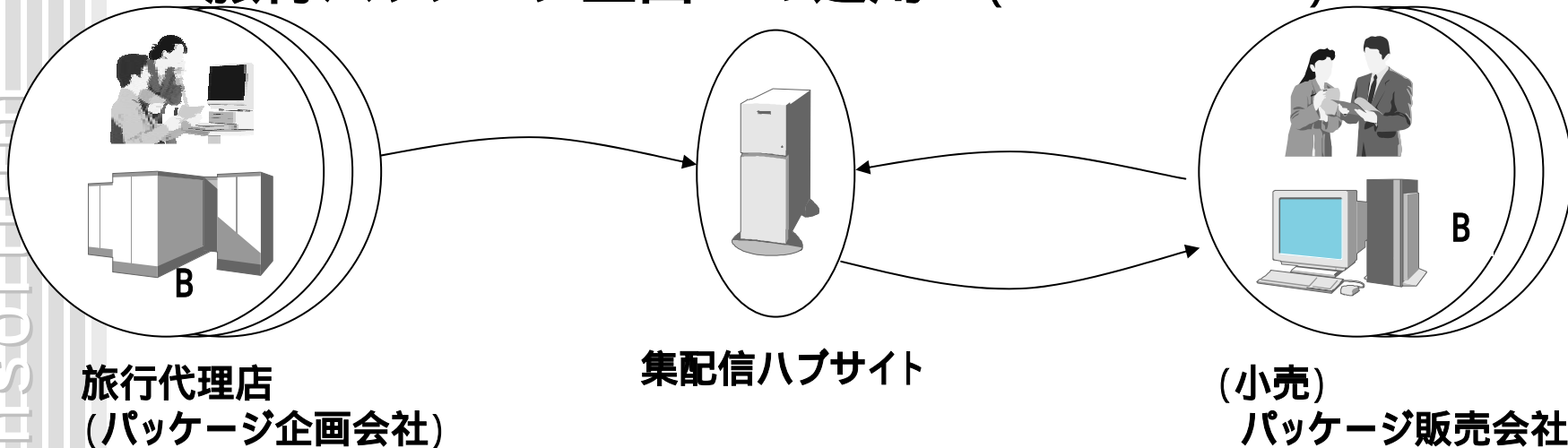


< 特長 >

1. データエントリー, ポータルサイト間の**情報流通をセリアルタイム**に行なうことができる。
2. 各サーバ間で流通する**データのフォーマットは任意に設定**することができる。
3. 同一システムで**B 2 B , G 2 B , B 2 C**など様々な運用が可能なシステムモデルである。
4. ポータルサイトは**必要な情報を自由に選択**することが可能である。
5. **システムの拡張が容易**(データエントリー、ポータルサイト)である。 (WSDLインタフェース公開)

2) 適用モデル例について

- 旅行パッケージ企画への適用 (TravelXML) -



< フロ - >

販売するパッケージツアーの登録を行なう。(金額, エリア, タイプなど)

小売はほしいパッケージの条件を事前設定する。(種別(海外), エリア, タイプ(自由旅行))
小売はほしいパッケージ商品の情報の入手することができる。

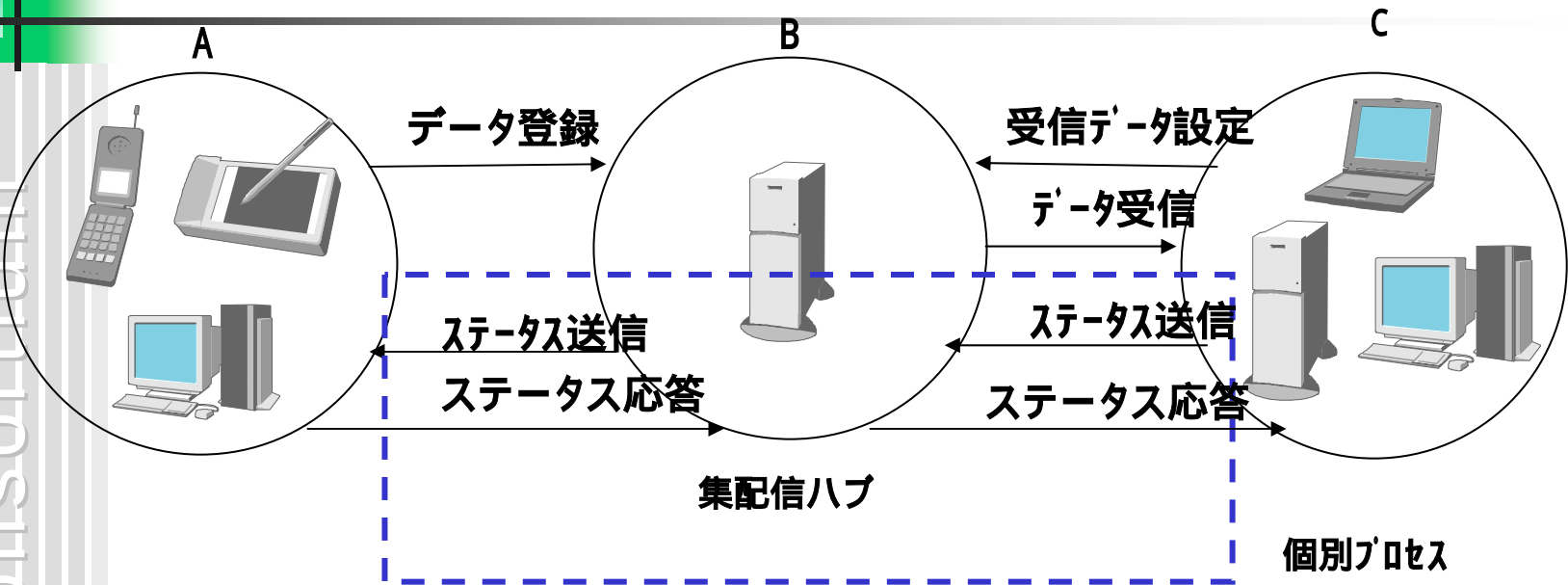
旅行代理店とホールセーラが契約交渉を行なう。

< メリット >

小売: 各社の**パッケージの比較を一度に行なう**ことができる。

旅行代理店: 自社パッケージを**新しい販路(小売)に紹介**することが可能になる。

- 旅行パッケージ企画への適用（相互接続モデル） -



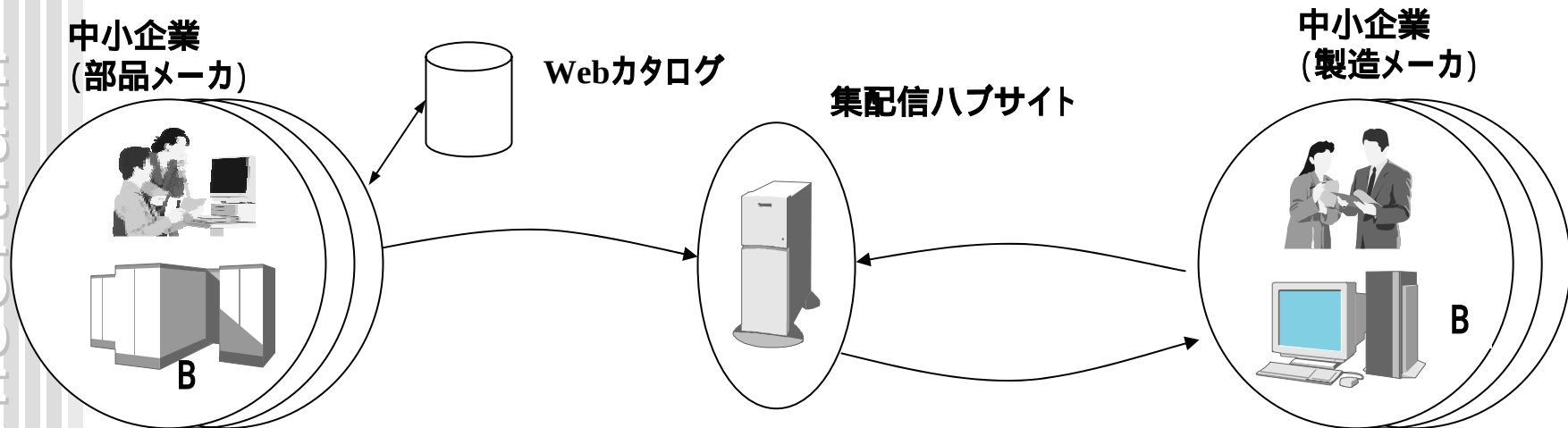
<フロー>

販売するパッケージツアーの登録を行なう。
個人利用者はほしいパッケージの条件を設定する。
パッケージ商品のデータを受信する
ステータスを送信する
ステータスを応答する

(A → B) [例: 種別 (国内, 国外), 価格]
(C → B) [例: 種別 (海外), 地域 (米国)]
(C → B)
(C → B → A) [売買条件 (金額)]
(A → B → C) [成約有無など]

3) 適用モデル例について

- 企業広報モデル (B2B) -



<フロー>

企業 (部品メーカー) が自社製品を登録する。

企業 (製造メーカー) は必要な商品の情報を登録する

中継ハブサイト経由で商品情報 (品名, 内容詳細, 価格など) が送られる。

<メリット>

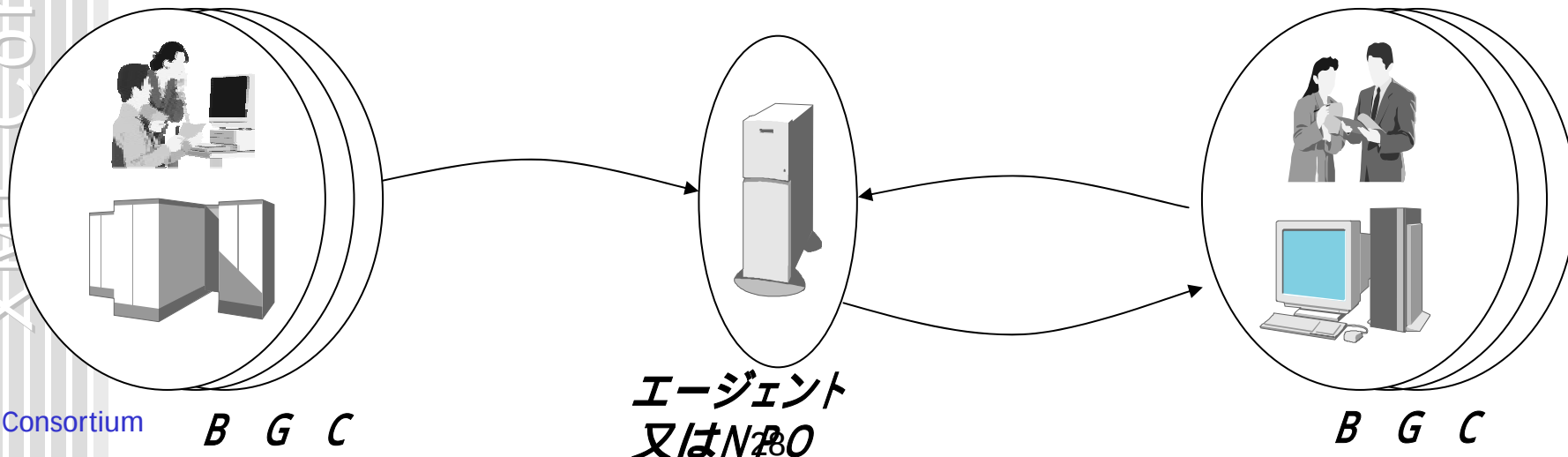
中小企業 (部品メーカー): 企業が販売したい商品の広報が可能。

中小企業 (製造メーカー): 自社が購入したい商品の広報情報の入手が可能。

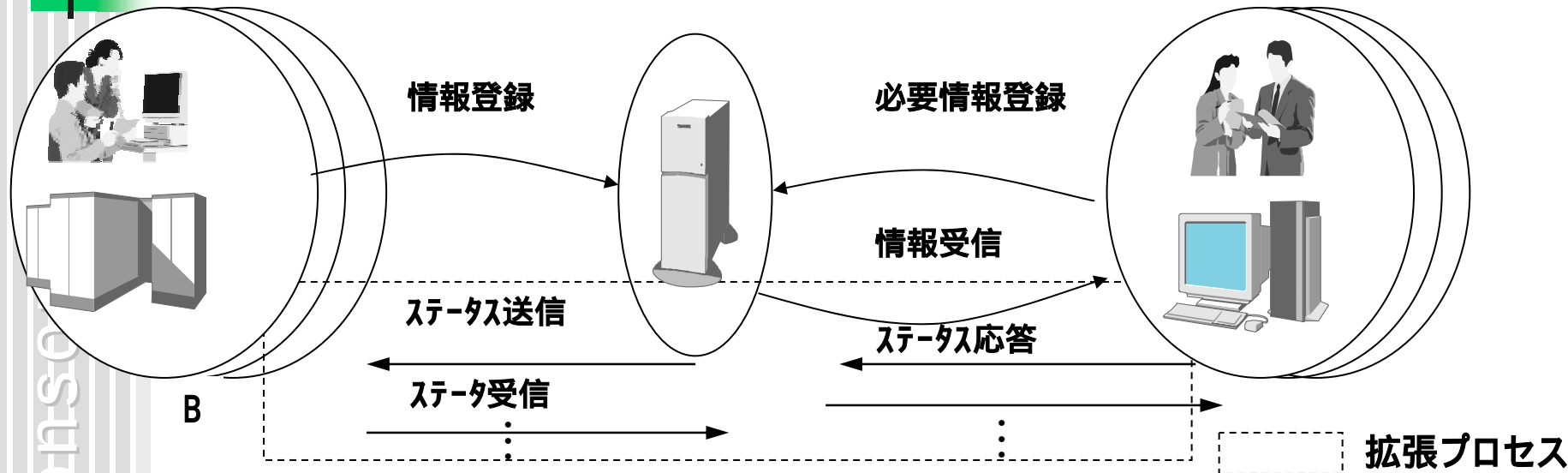
4) システムモデルのタイプ

このシステムモデルは提供する情報からの利益の有無により(NPOモデル), エージェントモデルに分類することができる。

- NPOモデル: 非営利団体型モデル。このモデルは収益を得ることを目的としないモデル 例: 企業広報サイト(業界団体運営)、政府広報
- エージェントモデル: 情報提供者, 情報利用者課の相互的なコミュニケーションにより収益を得ることを目的としたモデル 例: 旅行情報提供, 企業広報サイト, オークションサイト

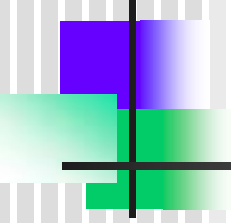


5) 実証実験システムモデルの評価



< 特長 >

- 1) **利用対象を選ばない (G, B, C) フレキシビリティなシステムモデルである。**
- 2) **プラットフォーム非依存、システム拡張の容易性、セミリアルタイム接続といった Web サービスの特長を活かしたシステムモデルである。**
- 3) **利用も目的や利益の有無により (NPOモデル, エージェントモデル) 使い分けることができるシステムモデルである。**
- 4) **基本プロセス (情報登録, 受信情報選択, 情報受信) に拡張プロセスを加えることにより幅広い利用が可能になる **拡張性の高いシステムモデル** である。**

A decorative graphic on the left side of the slide, consisting of a black crosshair overlaid on a blue square and a green square.

4 . システム開発の評価

1) Webサービス接続テスト回数

- WSDL策定前から参加
 - ポータル・サイト 19.5回(4社平均)
 - データ・エントリ 12回(1社)
- WSDL策定後から参加
 - ポータル・サイト 5回(1社)
 - データ・エントリ 3回(1社)



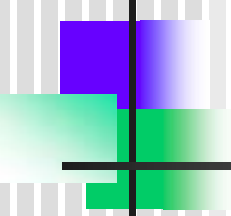
WSDL策定に大変な労力を必要とする

2) WSDL変更理由

- アプリケーション仕様に依存
 - 情報モデルの変更(施設情報, 季節情報の分離)
 - API引数の定義
 - データ型(long or int)
 - オプション項目扱い(空要素 or 要素なし)
 - データ定義と実データの不一致
 - 要素名スペルミス(綴り誤り, 大文字/小文字区別)
 - 名前空間の割り当てミス
- 製品実装に依存
 - クラス・バインディング(日本語要素名が不都合)
 - サービス・ディスパッチ方式(SOAPAction, Body直下要素の名前空間)
 - サーバ側WSDL取り込み機能
- アプリケーション仕様によるWSDL変更が大半！

3) 開発時間

- プロジェクト全体 1,343時間 9.6人月
(1月20日 1日7時間で計算)
- ポータルサイト(5社)
 - システム全体 1,024時間
 - インタフェース 135時間
 - 13.2%(インタフェース割合)
- 集配信(3社)
 - システム全体 299時間
 - インタフェース 12時間
 - 4.0%
- データエントリ(2社)
 - システム全体 90時間
 - インタフェース 15時間
 - 16.6%

A decorative graphic on the left side of the slide, consisting of a black crosshair overlaid on a purple square and a green square.

4) 開発者数

- プロジェクト全体 18人
- ポータルサイト(5社):11人
- 集配信(3社):5人
- データエントリ(2社):2人

5) 今回のWSDL策定のポイント

- タグ名を英語化
 - 日本語タグは自動プログラミングの障壁になる
- Doc/Literalを採用
 - ○ 相互接続性の確保
 - ○ 調整のしやすさ
 - 実装に手間
 - × 手作業で編集
 - XMLドキュメントからWSDLを作れますか!?
 - × ベンダーによって対応レベルに差異

SOAP 1.2 時代には大きく変化することを期待！

6) 使用プロダクト

<データ入力>

- Microsoft EXCEL 2000 , 2002
- Microsoft Office XP Web Services Toolkit 2.0
- Microsoft .NET Framework 1.0

<集配信サーバー>

- IBM WebSphere Application Server V4.05
- Oracle9i Application Server Release 2 (9.0.3)
- Windows Server 2003 Enterprise Edition RC2

<UDDIリポジトリ>

- Oracle9i Application Server Release 2 (9.0.3)

<データベース>

- Oracle9i Database Release 2 (9.2.0) : Oracle XML DB
- IBM DB2 UDD V8.1

<ポータルサーバー>

- NEC ActiveGlobe WebOTX Ver5.1
- Hitachi Cosminexus Application Server Version5
- Fujitsu Interstage Application Server V5.0L20
- Microsoft .NET Framework 1.0

<PDAポータルサーバー>

- Fujitsu Interstage Application Server V5.0L20

<携帯電話ポータルサーバー>

- NEC ActiveGlobe WebOTX Mobile Broker Ver1.1