



IBM Software Group

SCAによるメインフレームアプリケーションのSOA化 ～ CICS既存資産の活用例 ～

概要とメリット

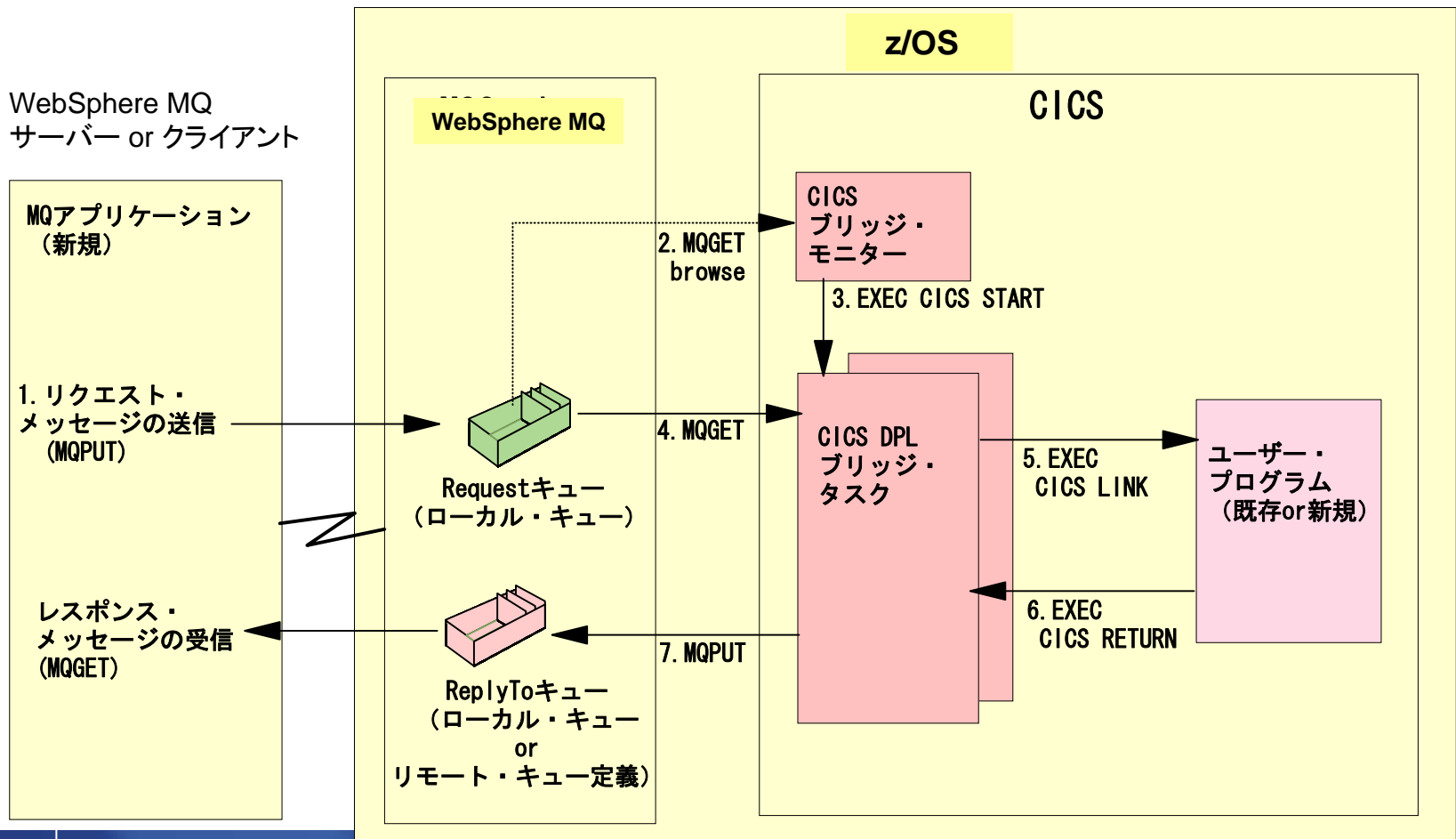
CICSをはじめとしたメインフレーム・アプリケーションをSOA対応しようという試みの一つをご紹介します。MQブリッジを使う場合、レガシー側の再構築が不要です。SCAという言葉が聞き慣れないかと思いますが、これは後ほどご説明します。

- MQ-CICSブリッジは昔から多くのお客様で使われています
- その仕組みを維持したまま、SCAとの接続を追加することによって、CICSアプリケーション活用の幅が大きく広がります
 - Webサービス
 - 他のSCAコンポーネント
 - Webアプリケーション
- 多くの部品開発が機械的にできるので、品質が安定します

(*) Service Component Architecture
SOAのIBM実装のひとつ

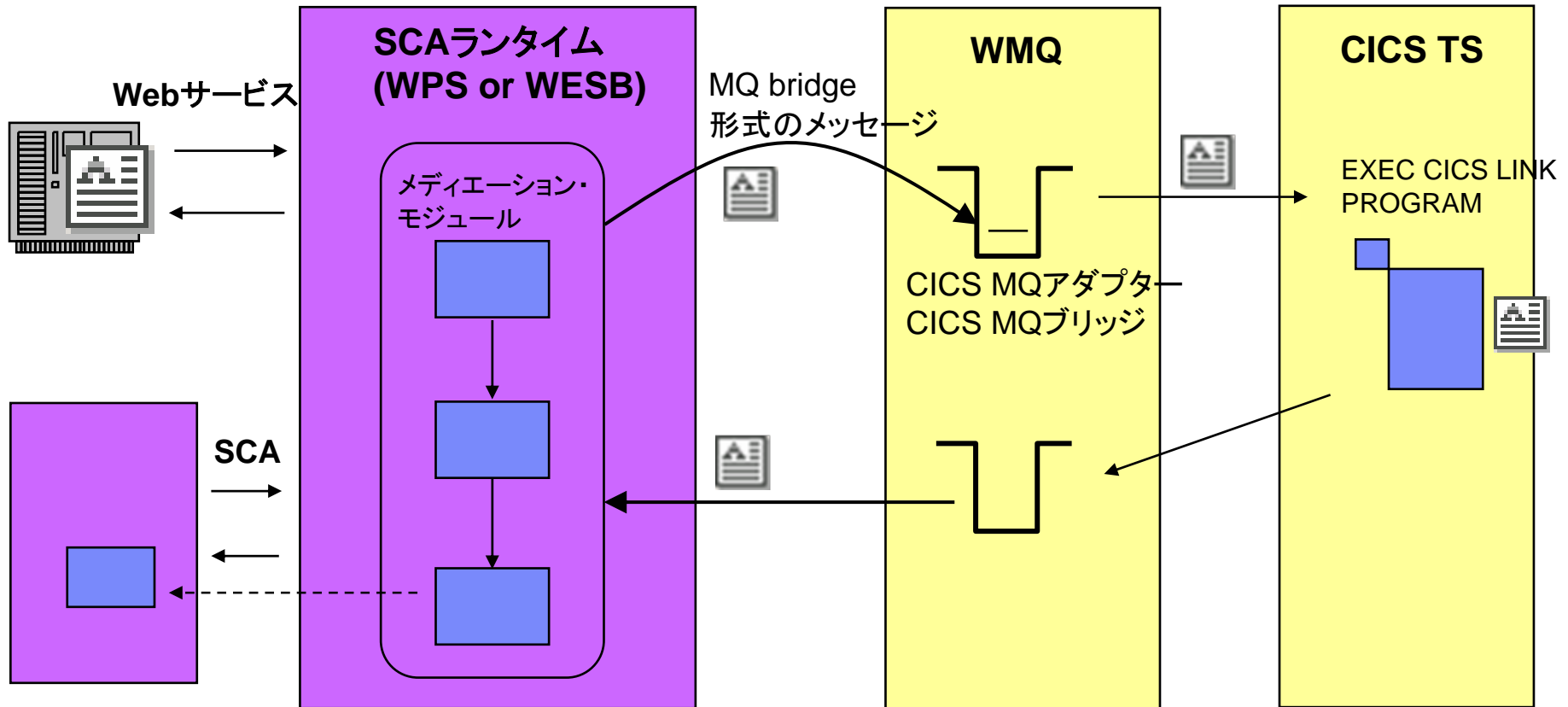
MQ-CICSブリッジ

CICSで古くから提供されているMQブリッジを使うと、CICSのアプリケーションをWebSphereMQ経由で呼び出すアプリケーションが簡単に構築できます。



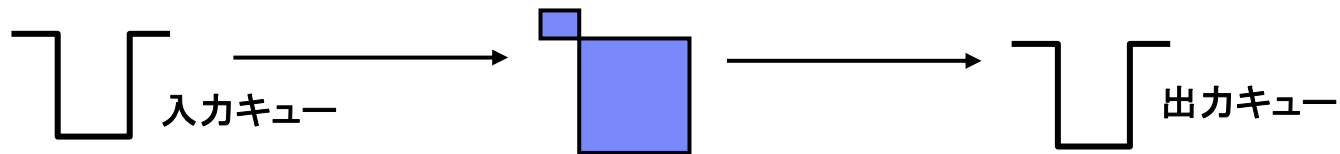
CICSのMQブリッジをSCAによりSOA対応に拡張する

SCAランタイムのサーバー層をフロントエンドに置くことで、Webサービスをはじめとした、SOA対応が簡単にできます。



メッセージ1： MQとクライアント間の送受信メッセージ

CICS-MQブリッジを介してCICSを呼び出すとき、MQとMQクライアント間で送受信するメッセージのフォーマットはあらかじめ規定されたMQメッセージ形式です。



```

StrucId : 'MD ' Version : 1
Report  : 0 MsgType : 1
Expiry  : -1 Feedback : 0
Encoding : 785 CodedCharSetId : 37
Format  : 'MQCICS '
Priority : 0 Persistence : 1
MsgId   : X'C3E2D840D4D8F6C240404040404040C325425CDCFB478B'
CorrelId : X'414D51214E45575F53455353494F4E5F434F5252454C4944'
BackoutCount : 0
ReplyToQ      : 'CICS.FROM.QUEUE'
ReplyToQMgr   : 'MQ6B'

00000000: C3C9 C840 0000 0002 0000 00B4 0000 0311 'CIH ..... '
00000010: 0000 0025 4040 4040 4040 4040 0000 0000 '..... '
00000020: 0000 0000 0000 0000 0000 0000 0000 0111 '..... '
00000030: FFFF FFFE 0000 0001 FFFF FFFF 0000 0000 '..... '
00000040: 0000 0000 0000 0000 0000 0000 0000 0000 '..... '
00000050: 0000 0000 4040 4040 4040 4040 4040 4040 '..... '
00000060: 4040 4040 4040 4040 4040 4040 4040 4040 '..... '
00000070: 4040 4040 4040 4040 4040 4040 4040 4040 '..... '
00000080: 4040 4040 4040 4040 4040 4040 4040 4040 '..... '
00000090: 4040 4040 4040 4040 4040 4040 4040 4040 '..... '
000000A0: 4040 4040 0000 0000 0000 0000 0000 0000 '..... '
000000B0: 0000 0000 C7C5 E3C5 D4D7 4040 40F0 F0F0 '....GETEMP 000'
000000C0: F1F0 F040 4040 4040 4040 4040 4040 4040 '100'
000000D0: 4040 4040 4040 4040 4040 4040 4040 4040 '..... '
000000E0: 4040 4040 4040 4040 4040 4040 4040 4040 '..... '
000000F0: 4040 4040 4040 4040 4040 4040 4040 4040 '..... '
00000100: 4040 4040 4040 4040 4040 4040 '..... '
  
```

CICS PGM

```

StrucId : 'MD ' Version : 1
Report  : 0 MsgType : 2
Expiry  : -1 Feedback : 0
Encoding : 785 CodedCharSetId : 500
Format  : 'MQCICS '
Priority : 0 Persistence : 1
MsgId   : X'C3E2D840D4D8F6C240404040404040C325448830F5428B'
CorrelId : X'C3E2D840D4D8F6C240404040404040C325448830F5428B'
BackoutCount : 0
ReplyToQ      : 'CICS.TO.QUEUE'
ReplyToQMgr   : 'MQ6B'

00000000: C3C9 C840 0000 0002 0000 00B4 0000 0311 'CIH ..... '
00000010: 0000 0025 4040 4040 4040 4040 0000 0000 '..... '
00000020: 0000 0000 0000 0000 0000 0000 0000 0111 '..... '
00000030: FFFF FFFE 0000 0001 FFFF FFFF 0000 0000 '..... '
00000040: 0000 0000 0000 0000 0000 0000 0000 0000 '..... '
00000050: 0000 0000 4040 4040 4040 4040 4040 4040 '..... '
00000060: 5C5C 5C5C 4040 4040 4040 4040 4040 4040 '..... ****'
00000070: 4040 4040 4040 4040 4040 4040 4040 4040 '..... '
00000080: 4040 4040 4040 4040 4040 4040 4040 4040 '..... '
00000090: 4040 4040 4040 4040 4040 4040 4040 4040 '..... '
000000A0: 4040 4040 0000 0000 0000 0000 0000 0000 '..... '
000000B0: 0000 0000 C7C5 E3C5 D4D7 4040 40F0 F0F0 '....GETEMP 000'
000000C0: F1F0 F0E2 4B40 C44B 40C2 D6D9 D4C1 D540 '100S. D. BORMAN '
000000D0: 4040 4040 4040 40E2 E4D9 D9C5 E86B 40C5 '..... SURREY, E'
000000E0: D5C7 D3C1 D5C4 4040 4040 40F3 F2F1 F5F6 'ENGLAND 32156'
000000F0: F7F7 F8F2 F640 F1F1 40F8 F15B F0F1 F0F0 '77826 11 81$0100'
00000100: 4BF1 F15C 5C5C 5C5C 5C5C 5C5C '..... .11***** '
  
```

メッセージ2: Webサービス (SOAP) メッセージ

Webサービスで使用されるSOAPメッセージ形式はXML形式のテキスト・データで、HTTPに乗せて送受信する形態が一般的です。

上り

```
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:q0="http://GETEMP"
  xmlns:q1="http://GETEMP/GETEMPInterface"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <q1:callGetemp>
      <input1>
        <q0:stat />
        <q0:numb>000100</q0:numb>
        <q0:name />
        <q0:addrx />
        <q0:phone />
        <q0:datex />
        <q0:amount />
        <q0:comment />
      </input1>
    </q1:callGetemp>
  </soapenv:Body>
</soapenv:Envelope>
```

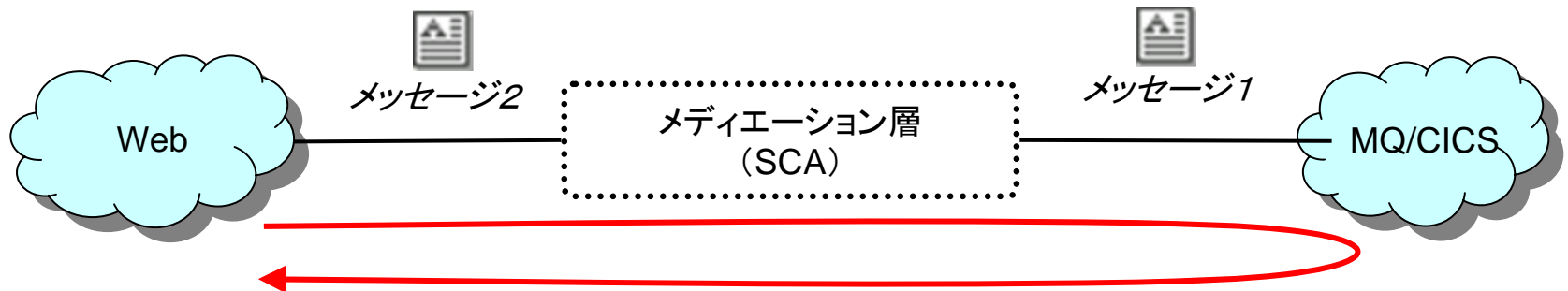
```
<soapenv:Envelope
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <mqs:MQMD
      xmlns:mqs="http://www.ibm.com/xmlns/prod/websphere/mq/sca/6.0.0">
      <Persistence>1</Persistence>
    </mqs:MQMD>
    <MsgId>
      C3E2D840D4D8F6C24040404040404040C3253FDF29421B4B
    </MsgId>
    <CorrelId>
      C3E2D840D4D8F6C24040404040404040C3253FDF29421B4B
    </CorrelId>
    <BackoutCount>0</BackoutCount>
    (20lines omitted)
  </mqs:MQMD>
  (20lines omitted)
  <mqs:MQHeaders
    xmlns:mqs="http://www.ibm.com/xmlns/prod/websphere/mq/sca/6.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <header>
      <Encoding>785</Encoding>
      <CodedCharSetId>500</CodedCharSetId>
      <Format>MQCICS</Format>
      <value xsi:type="mqs:MQCIH">
        (30 lines omitted)
      </value>
    </header>
  </mqs:MQHeaders>
</soapenv:Header>
<soapenv:Body>
  <in:callGetempResponse xmlns:ge="http://GETEMP"
    xmlns:in="http://GETEMP/GETEMPInterface">
    <output1>
      <ge:stat />
      <ge:numb>000100</ge:numb>
      <ge:name>S. D. BORMAN</ge:name>
      <ge:addrx>SURREY, ENGLAND</ge:addrx>
      <ge:phone>32156778</ge:phone>
      <ge:datex>26 11 81</ge:datex>
      <ge:amount>$0100.11</ge:amount>
      <ge:comment>*****</ge:comment>
    </output1>
  </in:callGetempResponse>
</soapenv:Body>
</soapenv:Envelope>
```

下り

つまりは メッセージ変換

このデモでは、SCAがメッセージ変換の役割を担っています。用語として、これをメディエーションと呼んでいます。

- MQメッセージとSOAPメッセージのフォーマット変換
- MQ通信とHTTP通信のプロトコル変換
- SCAの「メディエーション」層で、メッセージ変換を行っています



SCAって？

SCAはSOAのために用意された、コンポーネント化とその接続のための新技術です。

- SCA = Service Component Architecture
- IBMが提示するSOAの一つの実装
 - コンポーネント組み立て形式のアプリケーション構築が可能
 - 標準化も予定されている
- SCAにより以下のような各種技術に対し、単一のインターフェースでアクセス可能となる
 - EJB / Webサービス / Java / BPELビジネス・プロセスetc.
- SCAは「サービスの使用方法」と「サービスの開発方法」の双方をカバー

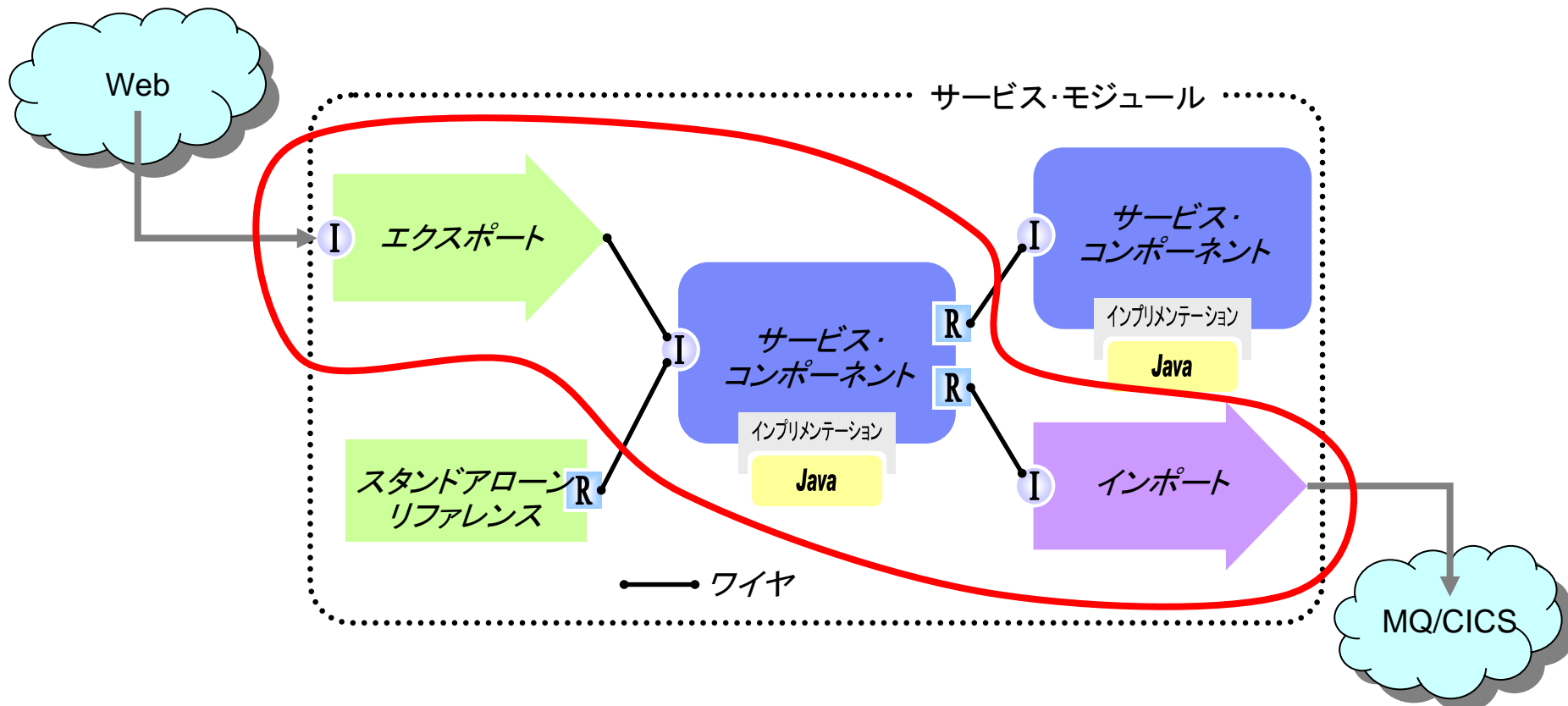
SCAの特徴

コンポーネント化とその接続には、プログラミング言語は関係ありません。ただしコンポーネントそのものの中身はJavaをはじめとした言語で開発します。

- コンポーネント組み立て形式のアプリケーション構築が可能。
 - プログラミング言語 (Java) を意識させない
 - WPS/WESB/WID上ではメディエーションやビジネス・プロセスなどは全てSCAコンポーネントとして扱われる
- 定義情報はSCDLというXMLファイルに保存される。
- コンポーネントのカスタマイズはコーディングではなく、GUI操作や各種のプロパティ設定にて可能。
- JavaやBPELなどの各種言語により、サービス・コンポーネントの実装を定義することが可能。
- インターフェースとしてJavaとWSDLをサポート

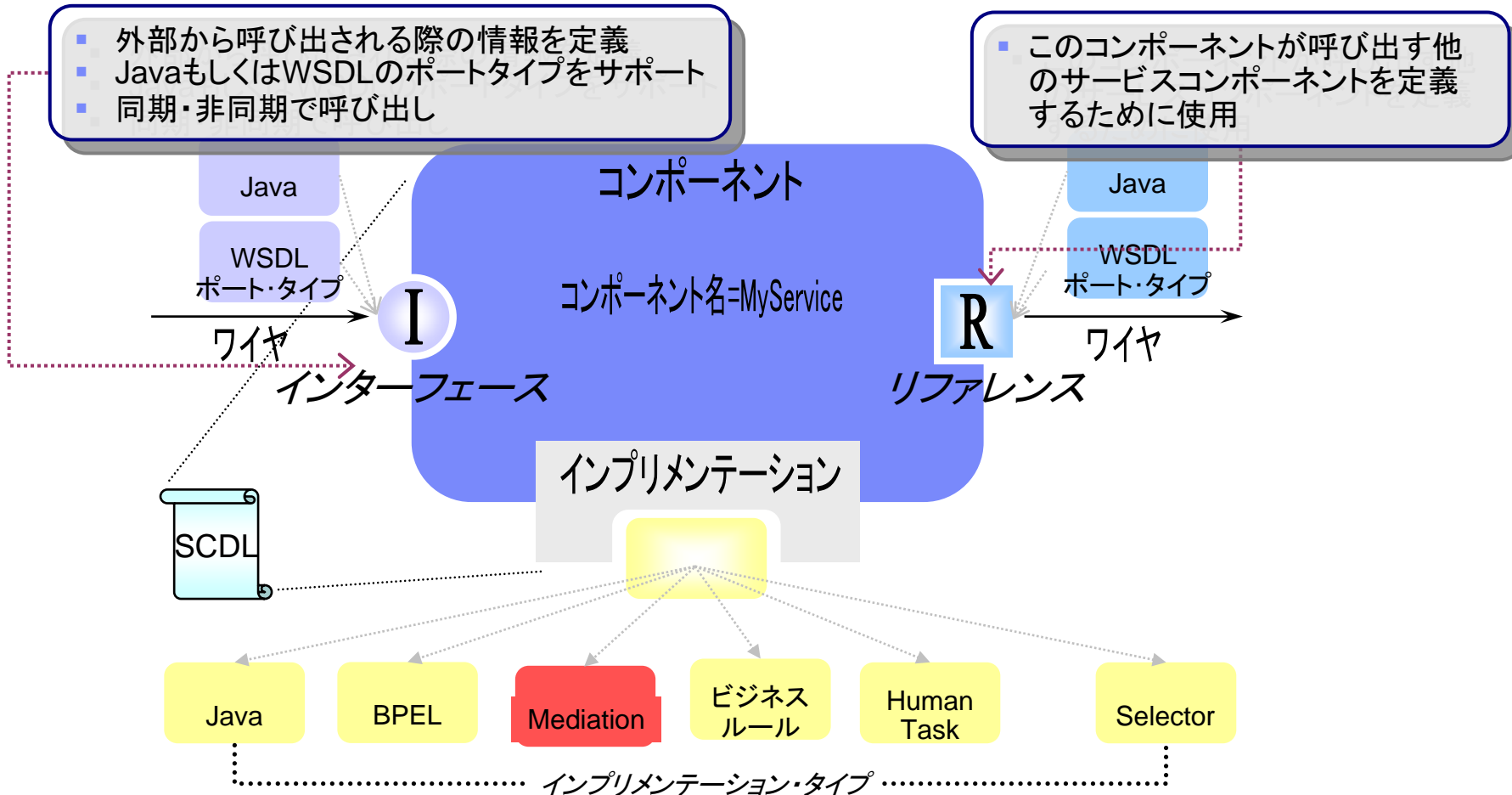
SCAのサービス・モジュール

SCAにおける、コンポーネント化の単位はサービス・モジュールです。他との接続のために、インポートとエクスポートというインターフェースがあるのが特徴です。



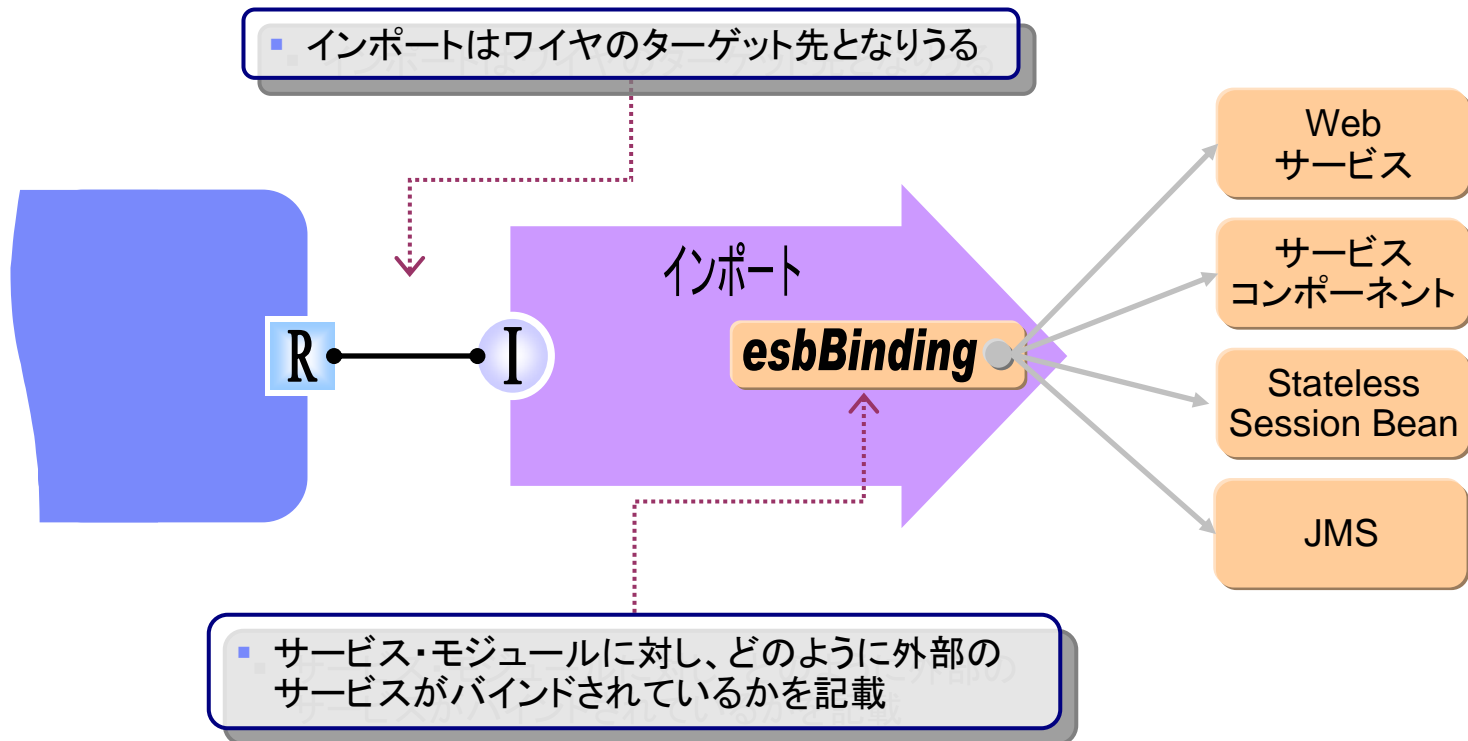
サービス・コンポーネント

コンポーネント化の核となるのが、実装を持つサービス・コンポーネントです。今回のデモでは、メディエーションの機能を使います。



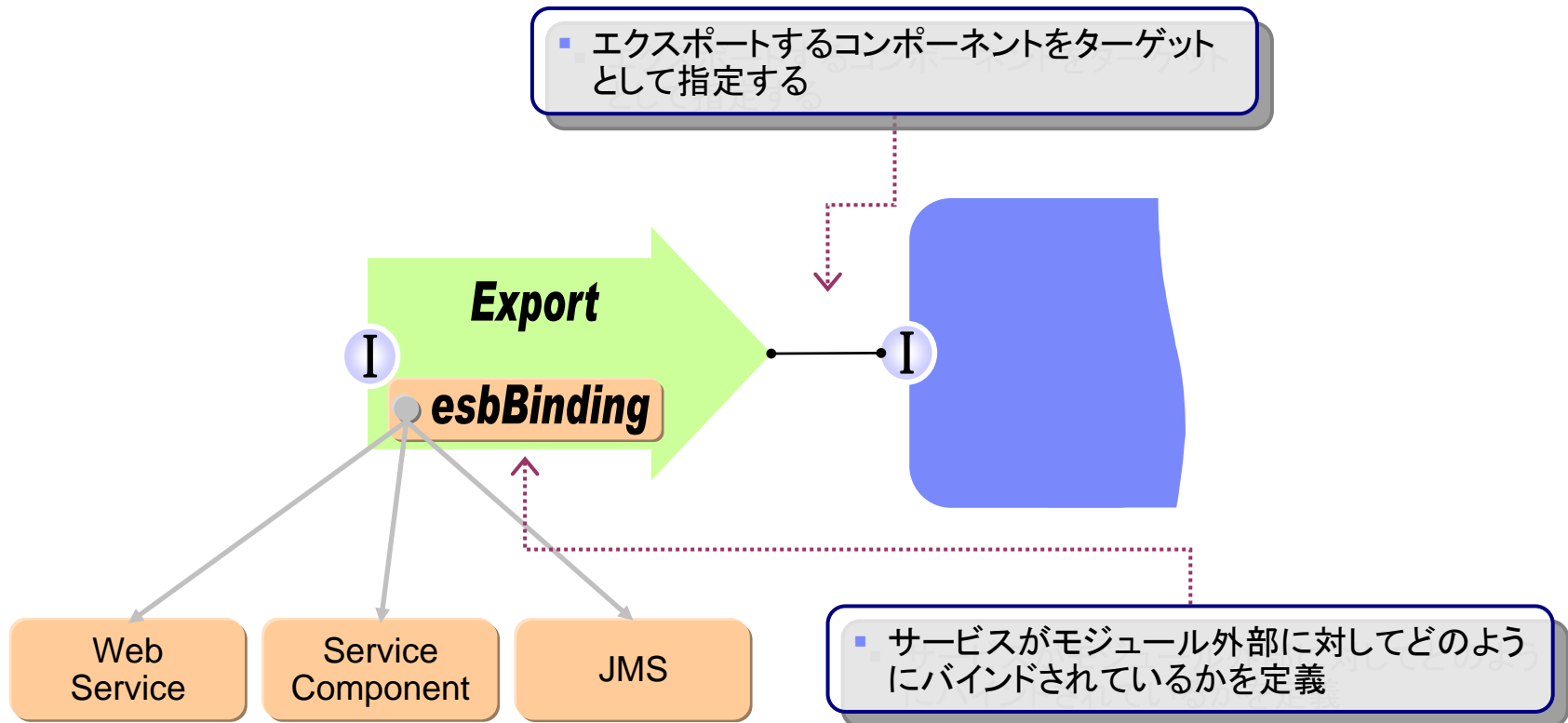
インポート

他のリソースからデータを取り込む、という意味で「インポート」と表現します。
また、そのデータへのアクセス方法を「バインディング」といいます。



エクスポート

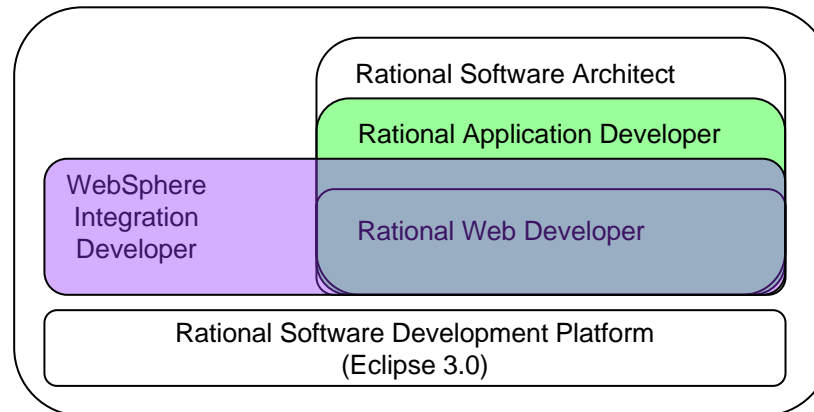
コンポーネントが、自分をどう公開するか、という意味で、「エクスポート」を作ります。インポートと同様にバインディングを持ちます。



開発環境(WebSphere Integration Developer)

SCAでコンポーネント開発をするためには、WIDというPC上のソフトウェアが必須です。もちろん、開発したモジュールはそのままz/OS上でも動きます。

- 全実装タイプのSCAモジュールを開発可能
 - WebSphere ESB上でメディエーション・モジュールが稼動可能
 - WebSphere Process Server (WPS)上で全実装タイプのモジュールが稼動可能
- Eclipseベースの統合開発環境



実行環境

SCAモジュールをプロダクション環境で動かすには、相応のサーバー製品が必要です。WPSやWESBはSCAに対応した弊社のサーバー製品です。



WebSphere Process Server (WPS)

ビジネスプロセスを設計し、自動化し、管理するというより上位のソリューションを提供

WebSphere ESBの機能を含む



WebSphere ESB (WESB)

ESBの機能を提供する新製品。
サービス間の接続とデータ変換機能を提供

WebSphere Application Server上に構築

WebSphere Application Server (WAS)

J2EEに準拠したアプリケーション・サーバー
最先端の可用性、拡張性、パフォーマンスを提供

**Increased
capabilities /
automation**



SCA実行環境

WebSphere Process Server (WPS)とは



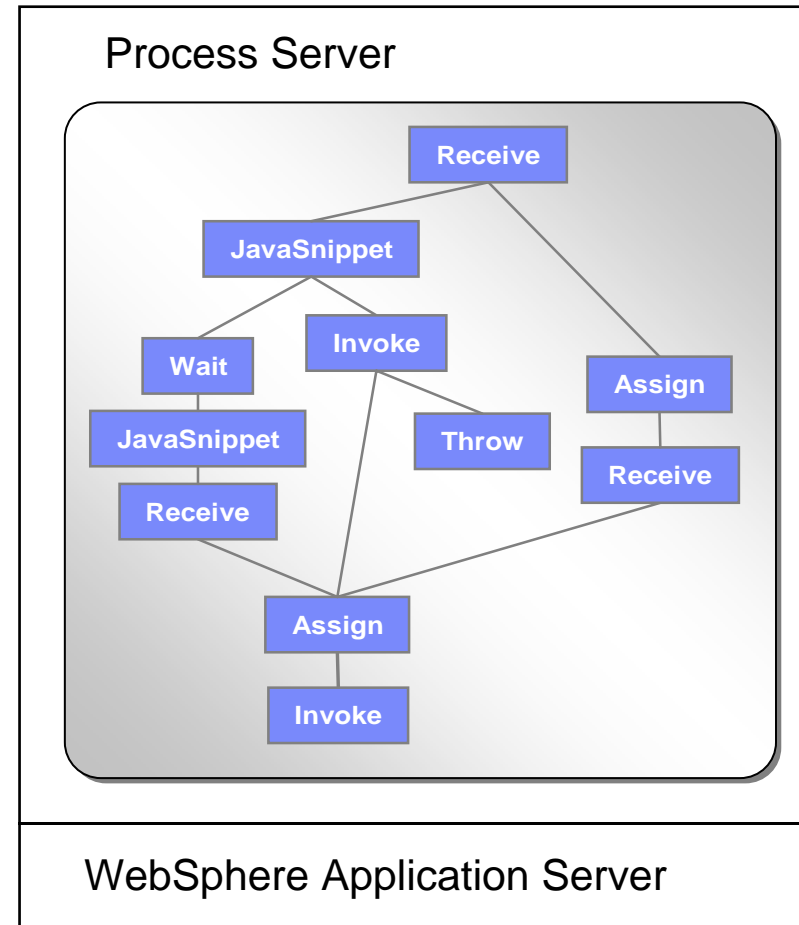
- WebSphere Application Server ベース
 - 堅牢かつ拡張性に富むプラットフォーム – クラスタ、フェイルオーバー、HA、セキュリティ
 - 統合管理環境 (Webベースの管理コンソール)
 - 共通イベント・インフラストラクチャー (CEI) – プロセス・マネジメント
- SOAプログラミング・モデル の実装
 - 統一プログラム呼び出しモデル (SCA)
 - 統一データ表記モデル (BO)
 - コンポーネントを作成・再利用するパワフルなツール
- ビジネス・プロセス・コレオグラファー
 - WS-BPEL 準拠
- 強化されたヒューマン・タスク
 - 様々なパターンの人による作業をタスクとして総合サポート・管理
 - 複数レベルのエスカレーション
 - サンプルのクライアント・コンポーネントを提供 (JSF)
- ビジネス・ステート・マシン、ビジネス・ルール、トランスフォーメーション
 - 統合ソリューションを構築するための拡張サービス
- 統一プラットフォームによる複雑性と管理コストの低減

ビジネス・プロセスとは

ビジネス・プロセス =
多くのアクティビティーやプロセスを機能的につ
なげたもの (Choreographer)

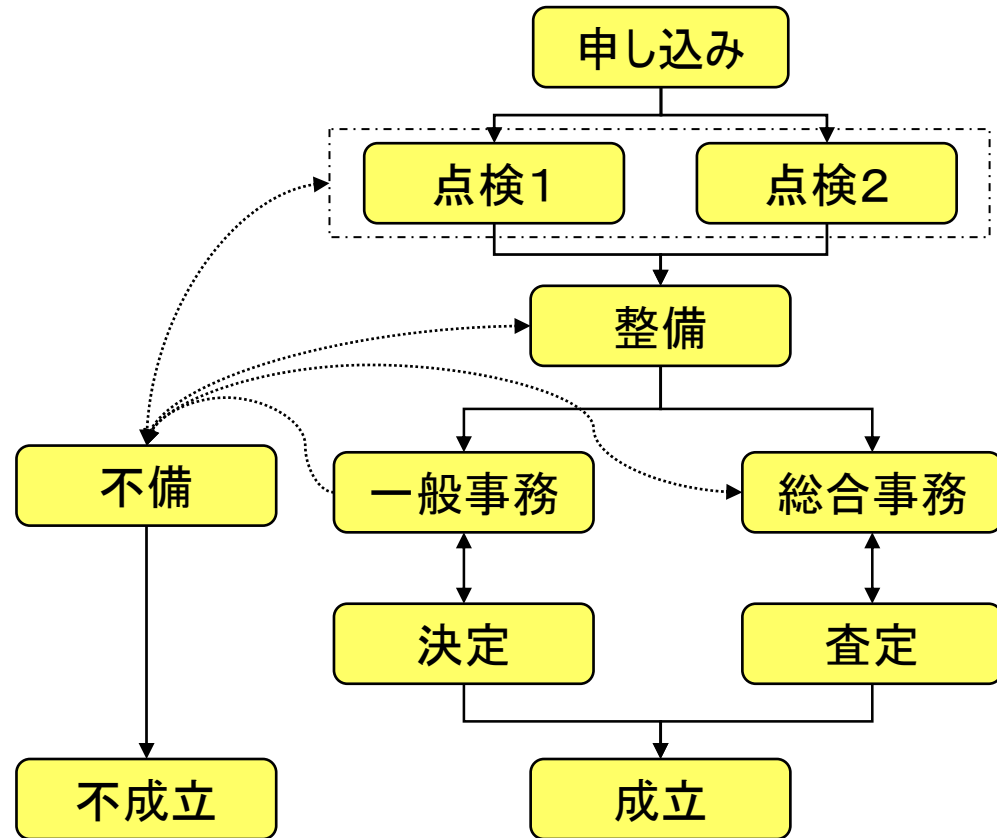
アクティビティー =
実際のビジネス・オペレーションを表したもの

- ✓ アクティビティーは、WSDLやJavaで定義される
- ✓ アクティビティー間の動作は、BPEL4WSで定義される



ビジネスプロセスの例： 損保申し込みプロセス

- シナリオ
 - 自動車保険新規申し込み
- ビジネス・フロー
 - 保険申し込み者が入力した内容をいくつかのプロセスを経て点検し、保険契約成立・不成立を決定する
 - 人手による意思決定を含むため、長期実行型のトランザクション



BPEL(4WS) 標準仕様

- BPEL は、ビジネス・プロセス・モデルを定義するための標準です
 - ✓ 実装に依存しないアプリケーション管理が可能になります
- WSDLをはじめとしたXML標準に基づいています
 - WSDL : Webサービスのインターフェース記述言語
 - XML Schema、XPath : データ・コンテキストやビジネス・ルールの表現
- IBM、Microsoft、BEA (& SAPとSiebel)によって提案されました
- OASIS (Organization for the Advancement of Structured Information Systems) による標準化

BPEL4WS (Business Process Execution Language For Web Services)

WSDL (Web Services Description Language)

XPath (XML Path Language)

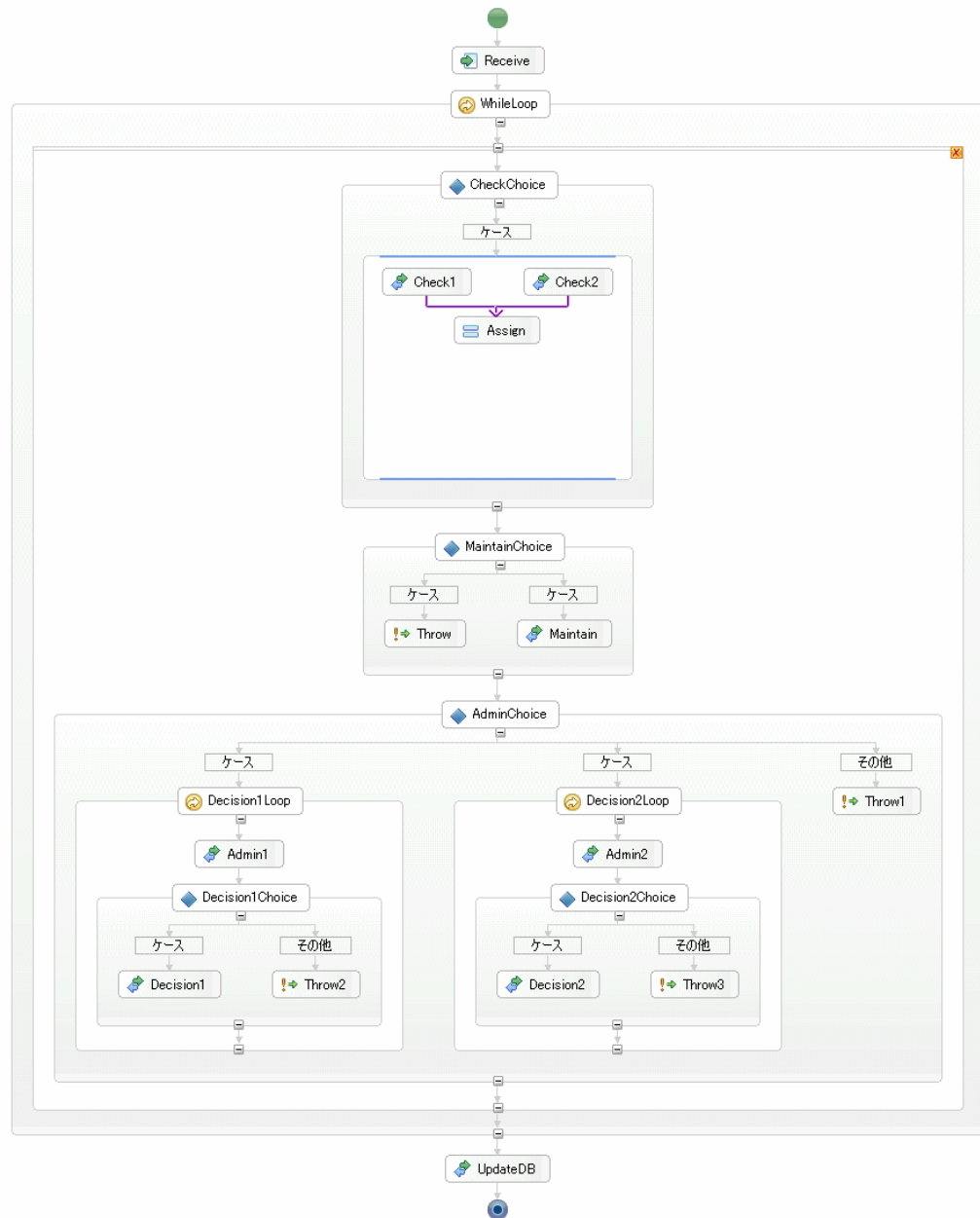
XSD (XML Schema)

XML

BPELの実装

左図は、開発ツールである WebSphere Integration Developer でのBPELの開発画面です。

GUIベースでの開発が可能となります



WS-BPEL 基本アクティビティー



Receive

Wait for a message to arrive. Optionally start a new process instance when the message arrives.



Reply

Reply to a message that was received.



Invoke

Invoke a one-way or a request-response operation offered by a partner.



Assign

Update the values of variables with new data.



Throw

Generate a fault from within the business process.



Rethrow

Throw an existing fault within a fault handler



Wait

Wait for a given period or until a certain time has passed.



Compensate

Invoke Compensation Handler



Terminate

Immediately terminate the process instance.



Empty

A "no-op" instruction in the business process.

WS-BPEL 構造化アクティビティー



Sequence

Multiple activities that are performed sequentially



Flow

Multiple activities that are performed concurrently



Switch (Choice)

Select one activity branch from a set of choices



Link

Synchronize two activities in a Flow to enforce a particular execution order



While

Repeat an activity until a boolean condition has been met



Pick (Receive Choice)

Receive and conditionally branch (convenient form of Receive combined with Switch)