

# PSLX技術仕様書

## XML標準規約 PSLX-04

PSLXコンソーシアム

<http://www.pslx.org>

# もくじ

- はじめに
- パート1:XMLタグ仕様
- パート1:タグ表記サンプル
- パート1:照会／集計の表現
- パート2:PSLXメッセージ仕様
- パート2:メッセージ交換サンプル
- パート3:PSLX通信規約
- パート3:ミドルウェアとのバインド方法
- まとめ

# 本仕様書の目的

APSエージェントとなる業務アプリケーションプログラムが、自分自身あるいは他のエージェントとの間で情報交換を行う際の、XML記述に関する規約と、そのデータを複数の業務アプリケーションプログラム間で交換するためのインタフェースの規約を定めることを目的とする。

# 仕様書のスコープ

レイヤー1

製造業ビジネスモデル層

レイヤー2

APSコラボレーション層

レイヤー3

APSエージェント層

レイヤー4

ドメインオブジェクト層

レイヤー5

データインプリメント層

# 他の仕様書との関係

- ドメインオブジェクトとの関係

- XMLのスキーマは、PSLXドメインオブジェクトのひとつの実装形態である。個別スキーマ生成規則による変換マップにより互換性を保証。

- APSエージェントモデルとの関係

- XMLのメッセージ仕様は、APSエージェントモデルの基本エージェントがもつべきインタフェースをすべて持つ。データ内容も準拠。

# 仕様書の構成

- パート1
  - XMLタグの記述方法(タグ記述規約)
- パート2
  - PSLXメッセージの仕様(メッセージ規約)
- パート3
  - 通信ミドルウェア利用方法(通信規約)

# もくじ

- はじめに
- **パート1：XMLタグ仕様**
- パート1：タグ表記サンプル
- パート1：照会／集計の表現
- パート2：PSLXメッセージ仕様
- パート2：メッセージ交換サンプル
- パート3：PSLX通信規約
- パート3：ミドルウェアとのバインド方法
- まとめ

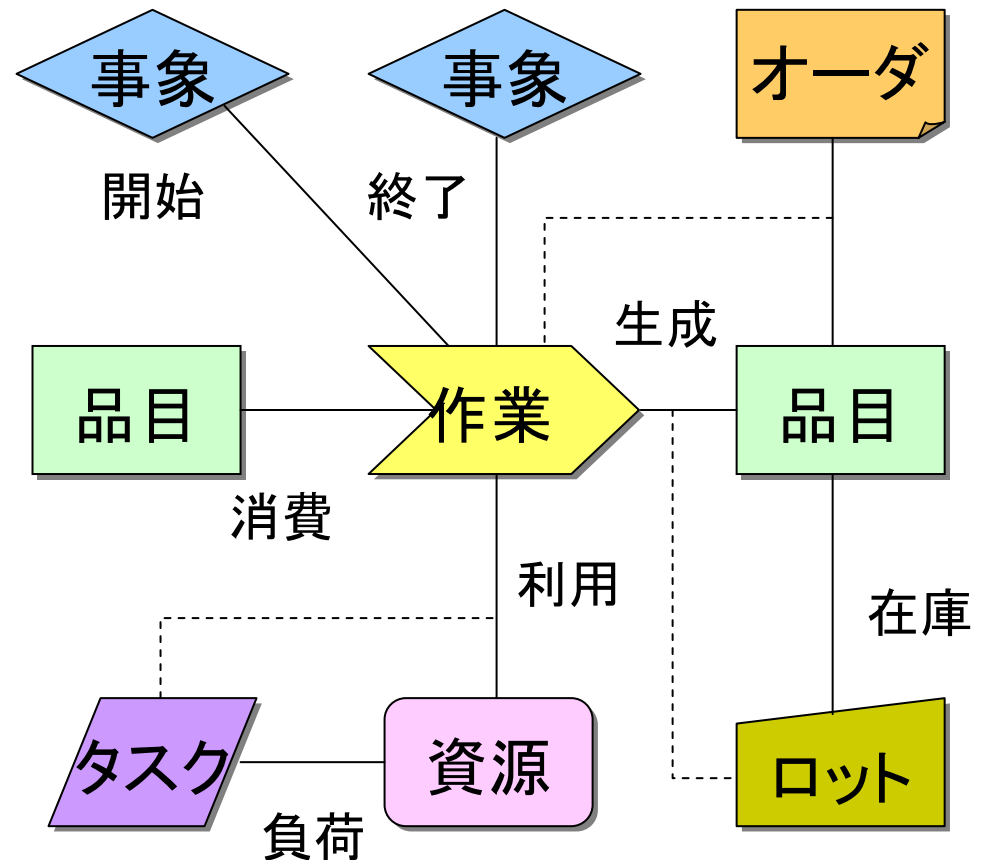
# タグ設計の基本方針

- メッセージは業務のスナップショット
- プロセスによるデータの多義性の対応
- オントロジーから業務用語への対応
- とにかくシンプルに、汎用的で美しく！



# 基本となる構成要素

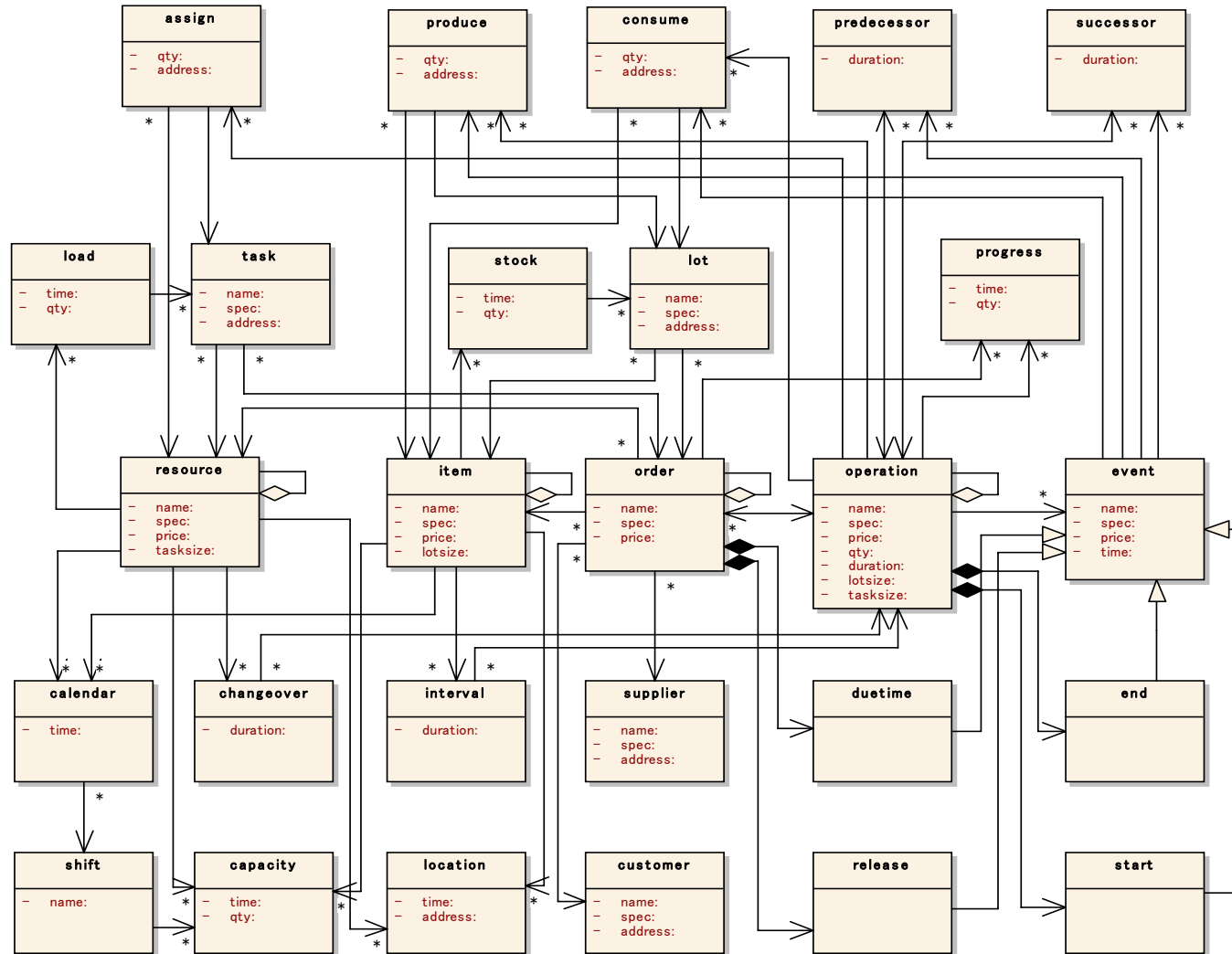
- 品目<item>
- 資源<resource>
- 作業<operation>
- 事象<event>
- ロット<lot>
- タスク<task>
- オーダ<order>



# PSLXタグ一覧

<pslx>	<char>	<produce>	<event>	<expression>
<profile>	<address>	<consume>	<ev>	<op>
<error>	<description>	<assign>	<start>	<parameter>
<color>	<time>	<predecessor>	<end>	<query>
<display>	<duration>	<successor>	<release>	<min>
<unit>	<spec>	<partof>	<duetime>	<max>
<translate>	<location>	<pegging>	<customer>	<earliest>
<scale>	<progress>	<tracking>	<supplier>	<latest>
<qty>	<capacity>	<lotsize>	<item>	<shortest>
<price>	<load>	<tasksize>	<resource>	<longest>
<base>	<stock>	<condition>	<lot>	<enumerate>
<priority>	<shift>	<action>	<task>	
	<calendar>	<changeover>	<operation>	
		<interval>	<order>	

# XMLスキーマ



# もくじ

- はじめに
- パート1:XMLタグ仕様
- **パート1:タグ表記サンプル**
- パート1:照会／集計の表現
- パート2:PSLXメッセージ仕様
- パート2:メッセージ交換サンプル
- パート3:PSLX通信規約
- パート3:ミドルウェアとのバインド方法
- まとめ

# 基本情報の表現

## 数値情報の表現

```
<qty value="100" unit="個"/>
```

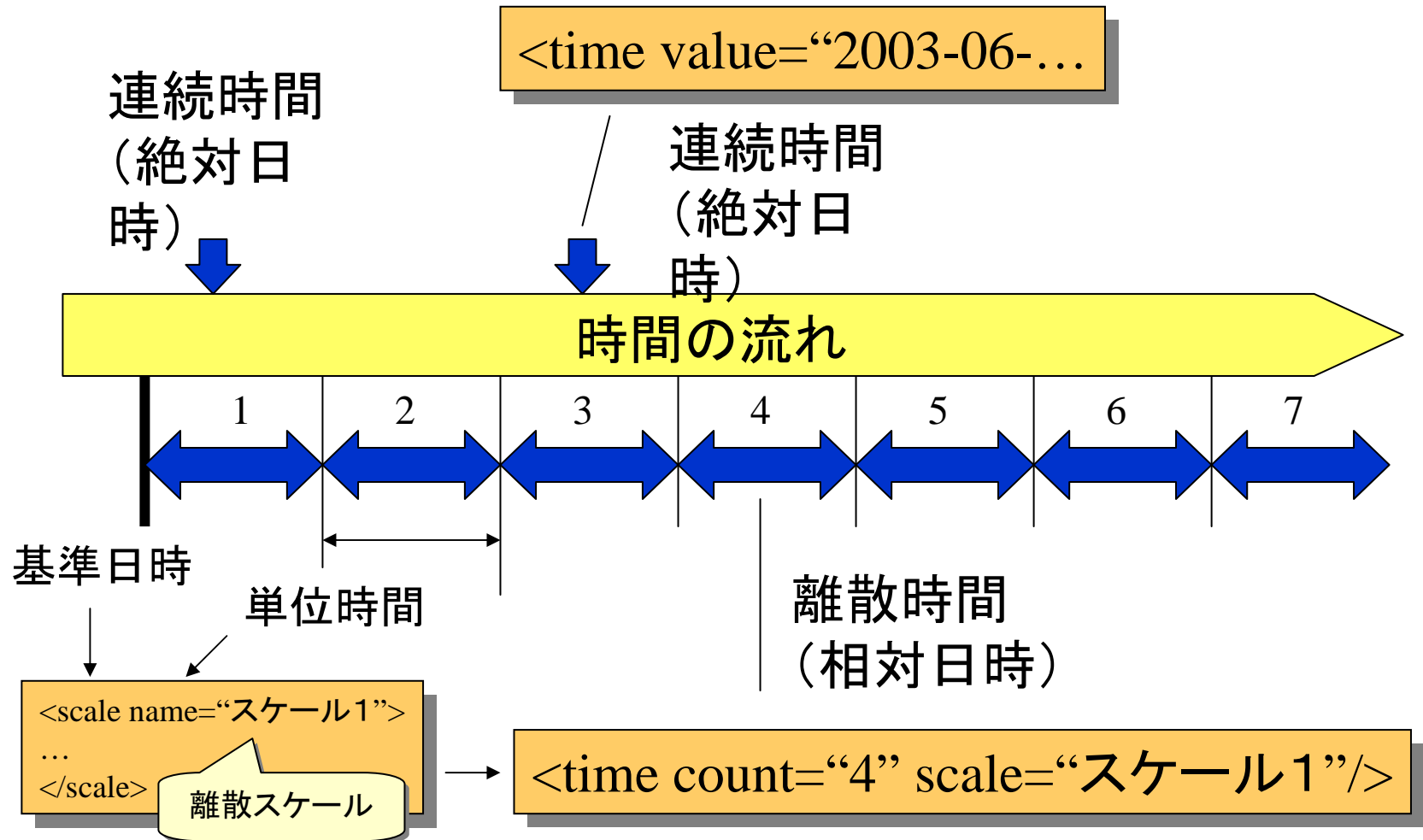
## 文字情報の表現

```
<char value="正常"/>
```

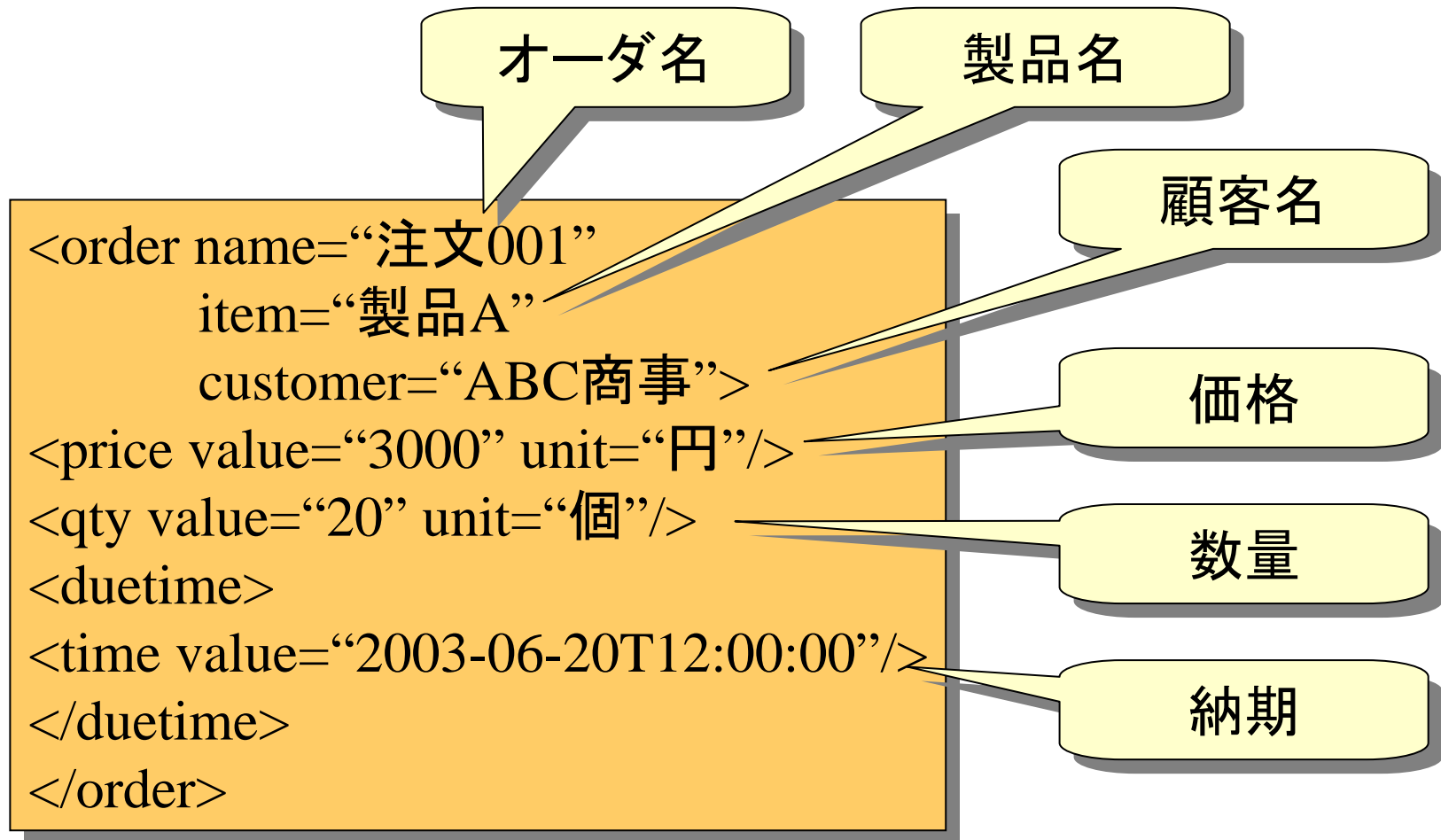
## 日時の表現

```
<time value="2003-06-18T13:00:00.000"/>
```

# 連続時間と離散時間



# オーダーの表現



# 進捗報告

## オーダーの進捗状況

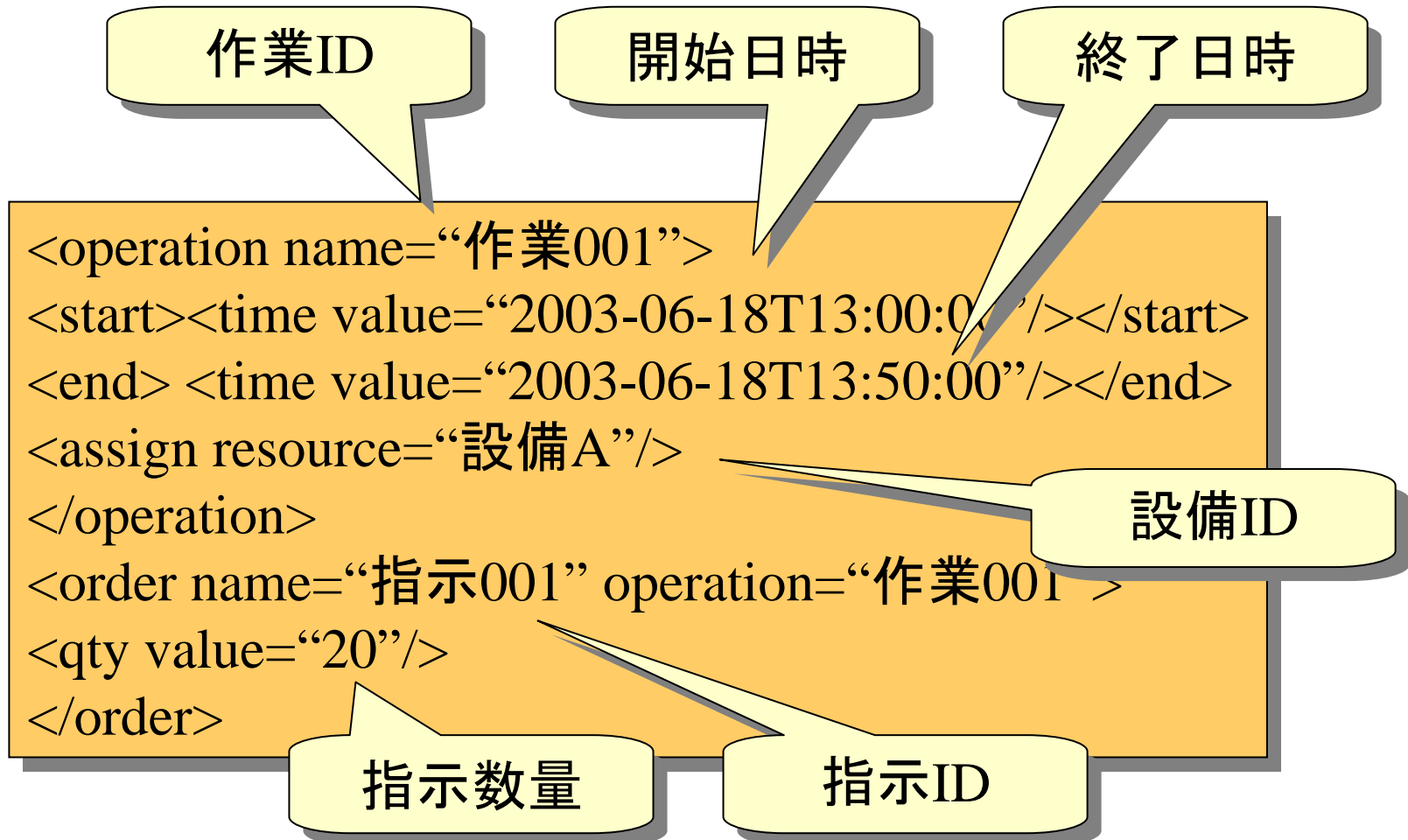
```
<order name="注文001">  
<progress><qty value="80" unit="%"/></progress>  
</order>
```

## 作業の進捗状況

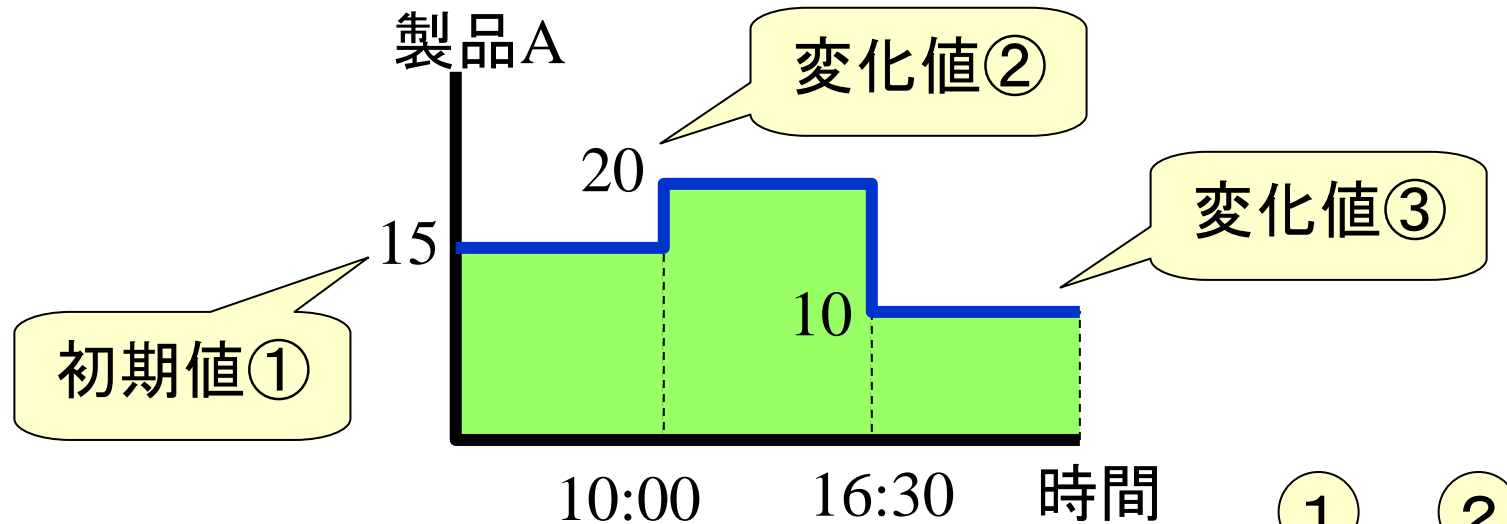
```
<operation name="注文001">  
<progress status="started">  
<qty value="150" unit="個"/>  
</progress>  
</operation>
```



# 作業指示



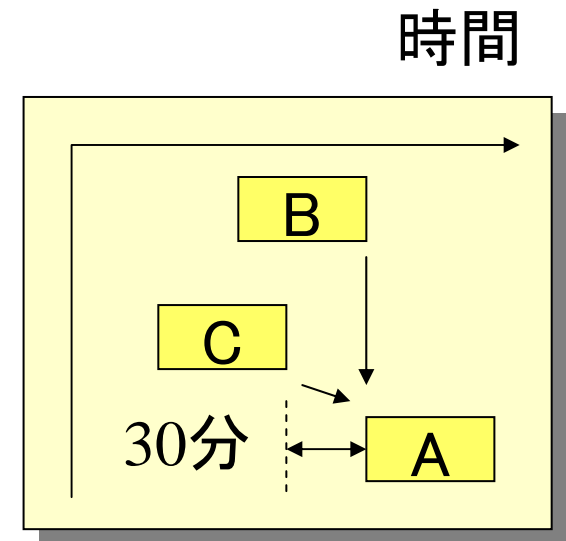
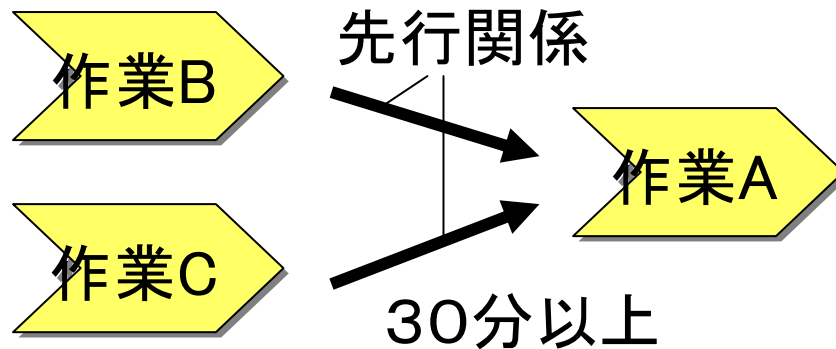
# 在庫／負荷の照会



```
<item name="製品A">  
<stock><time ref="init"/><qty value="15"/></stock>  
<stock><time value="2003-06-18T10:00:00"/><qty value="20"/></stock>  
<stock><time value="2003-06-18T16:30:00"/><qty value="10"/></stock>  
</item>
```

# 先行関係（作業マスター）

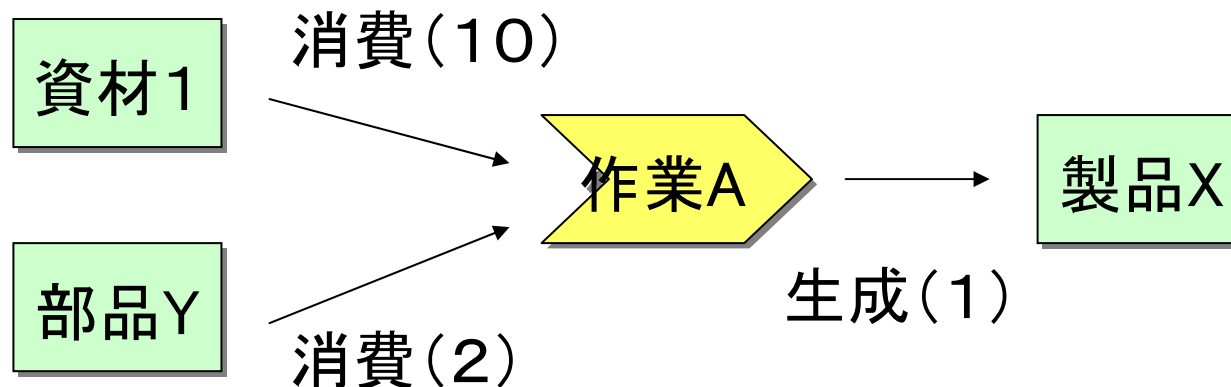
```
<operation name="作業A">  
<predecessor operation="作業B"/>  
<predecessor operation="作業C">  
  <duration value="PT30M"/>  
</predecessor>  
</operation>
```



スケジュール例

# 品目の生成・消費（作業マスター）

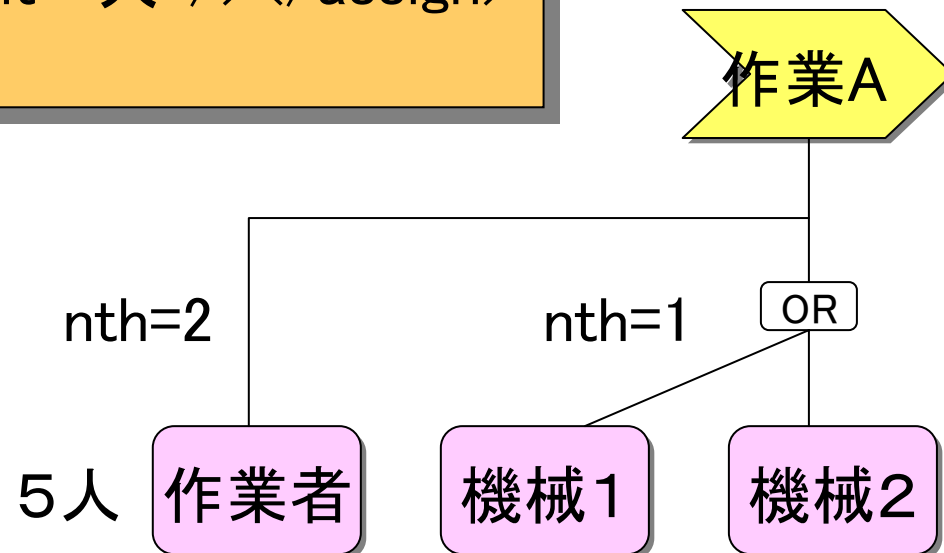
```
<operation name=“作業A”>  
<consume item=“資材1”><qty value=“10”/></consume>  
<consume item=“部品Y”><qty value=“2”/></consume>  
<produce item=“製品X”><qty value=“1”/></produce>  
</operation>
```



# 資源の割当（作業マスター）

```
<operation name="作業A">  
<assign resource="機械1" nth="1"/>  
<assign resource="機械2" nth="1"/>  
<assign resource="作業者" nth="2">  
<qty value="5" unit="人"/></assign>  
</operation>
```

} 代替資源



# もくじ

- はじめに
- パート1:XMLタグ仕様
- パート1:タグ表記サンプル
- **パート1:照会／集計の表現**
- パート2:PSLXメッセージ仕様
- パート2:メッセージ交換サンプル
- パート3:PSLX通信規約
- パート3:ミドルウェアとのバインド方法
- まとめ

# データの照会方法

サイズが130以上の製品名を  
価格とともにリストアップせよ。

```
SELECT name, price FROM item  
WHERE size ≥ 130
```



SQL表記

品目テーブル(item)

name	size	price
製品A	150	2000
製品B	100	2100
製品C	190	2500

```
<item name="#query">  
<spec name="size"><qty><min value="130"/></qty></spec>  
<query select="price"/>  
</item>
```

PSLX表記



```
<item name="製品A">  
<price value="2000"/></item>  
<item name="製品C">  
<price value="2500"/></item>
```

# データの照会方法

照会対象を条件で指定する場合には#queryとする。

照会したいデータ項目を指定する。  
nameは常に返る。

<min>、<max>等を用いて値の範囲指定が可能。

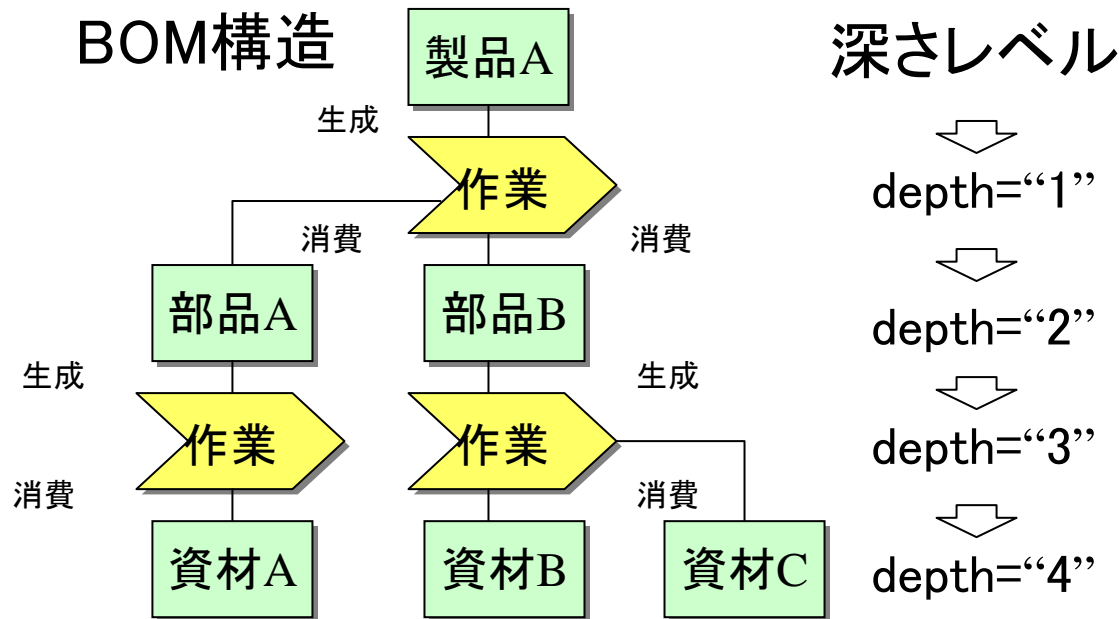
```
<item name="#query">  
<query select="price"/>  
<spec name="size"><qty><min value="130"/></qty></spec>  
</item>
```

タグの中に書かれた情報は、すべて制約条件とみなす。



# 階層構造の照会

```
<item name="製品A">  
<query select="produce" depth="4"/>  
</item>
```



その他の階層構造の例

ロット追跡  
先行関係  
紐付け関係

# データの集計方法

サイズが130以上の製品の平均価格を知りたい。

平均値の計算

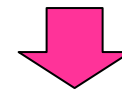
```
<item name="#query">  
<query select="price" calculation="ave"/>  
<spec name="size"><qty><min value="130"/></qty></spec>  
</item>
```

品目テーブル (item)

name	size	price
製品A	150	2000
製品B	100	2100
製品C	190	2500

①条件で  
絞込み

②平均  
を求める



```
<item name="#query">  
<price value="2250"/>  
</item>
```

# もくじ

- はじめに
- パート1：XMLタグ仕様
- パート1：タグ表記サンプル
- パート1：照会／集計の表現
- **パート2：PSLXメッセージ仕様**
- パート2：メッセージ交換サンプル
- パート3：PSLX通信規約
- パート3：ミドルウェアとのバインド方法
- まとめ

# メッセージ一覧

initPlan	計画を初期化する	setEstimation	オーダ見積を設定する
makePlan	計画を作成する	getEstimation	オーダ見積を照会する
initSchedule	スケジュールを初期化する	setProgress	進捗情報を設定する
makeSchedule	スケジュールを作成する	getProgress	進捗情報を照会する
setCalculation	計算方法を設定する	setStock	在庫情報を設定する
getCalculation	計算方法を照会する	getStock	在庫情報を照会する
setPlan	計画を設定する	setLoad	負荷情報を設定する
getPlan	計画を照会する	getLoad	負荷情報を照会する
setSchedule	スケジュールを設定する	setCapacity	能力情報を設定する
getSchedule	スケジュールを照会する	getCapacity	能力情報を照会する
setParty	顧客／サプライヤ情報を登録する	setCalendar	カレンダー情報を設定する
getParty	顧客／サプライヤ情報を照会する	getCalendar	カレンダー情報を照会する
setProduct	製品マスター情報を登録する	setLot	ロット情報を設定する
getProduct	製品マスター情報を照会する	getLot	ロット情報を照会する
setProcess	工程マスター情報を登録する	setTask	タスク情報を設定する
getProcess	工程マスター情報を照会する	getTask	タスク情報を照会する
setOrder	オーダ情報を設定する		
getOrder	オーダ情報を照会する		

# メッセージの種類

- 要求メッセージ

- クライアントがサーバーに何らかの処理を依頼するためのメッセージ

- 応答メッセージ

- サーバーがクライアントからの要求に対応するためのメッセージ

- 受領メッセージ

- 要求または応答メッセージを受領したことを伝えるためのメッセージ

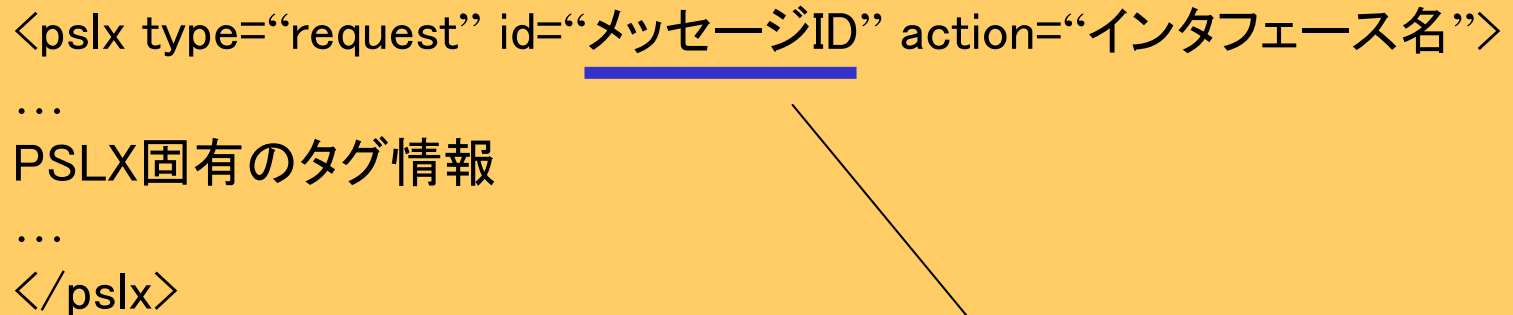
- 例外メッセージ

- 要求または応答メッセージに例外があることを伝えるためのメッセージ

# メッセージの基本形

## 要求メッセージ

```
<pslx type="request" id="メッセージID" action="インタフェース名">  
...  
PSLX固有のタグ情報  
...  
</pslx>
```



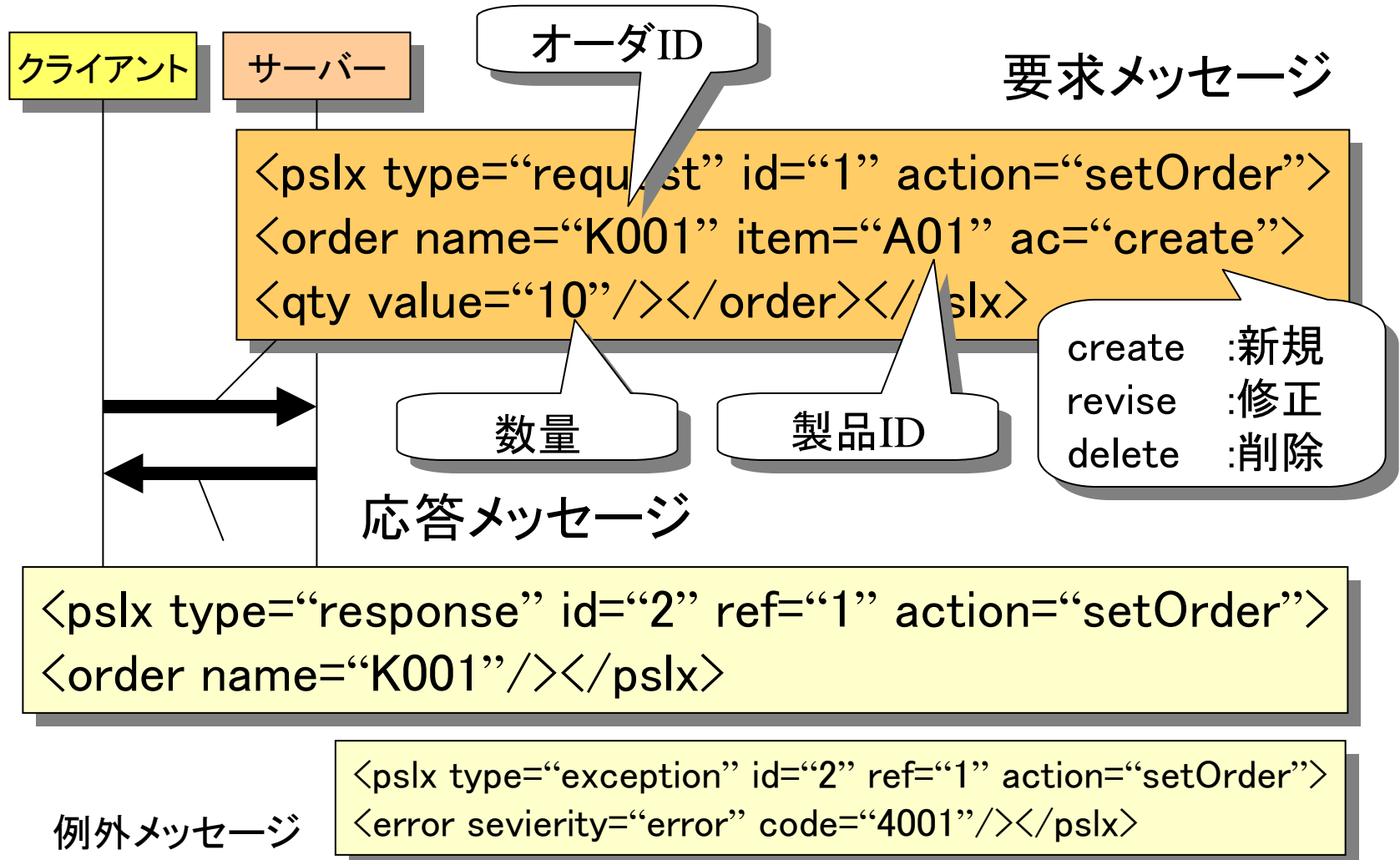
## 応答メッセージ

```
<pslx type="response" id="メッセージID" ref="参照メッセージID"  
action="インタフェース名">  
...  
PSLX固有のタグ情報  
...  
</pslx>
```

# もくじ

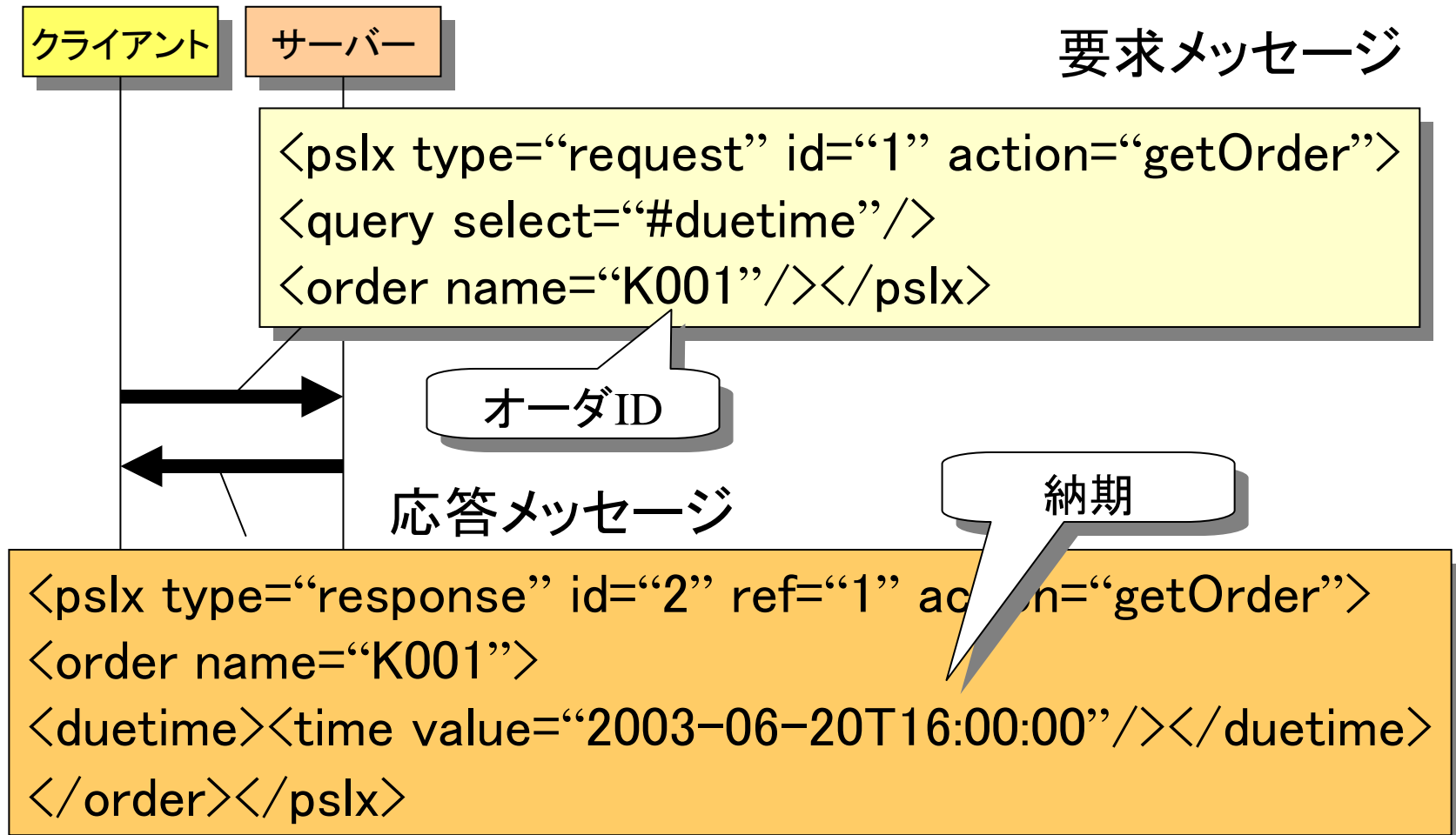
- はじめに
- パート1：XMLタグ仕様
- パート1：タグ表記サンプル
- パート1：照会／集計の表現
- パート2：PSLXメッセージ仕様
- **パート2：メッセージ交換サンプル**
- パート3：PSLX通信規約
- パート3：ミドルウェアとのバインド方法
- まとめ

# オーダー登録 (setOrder)

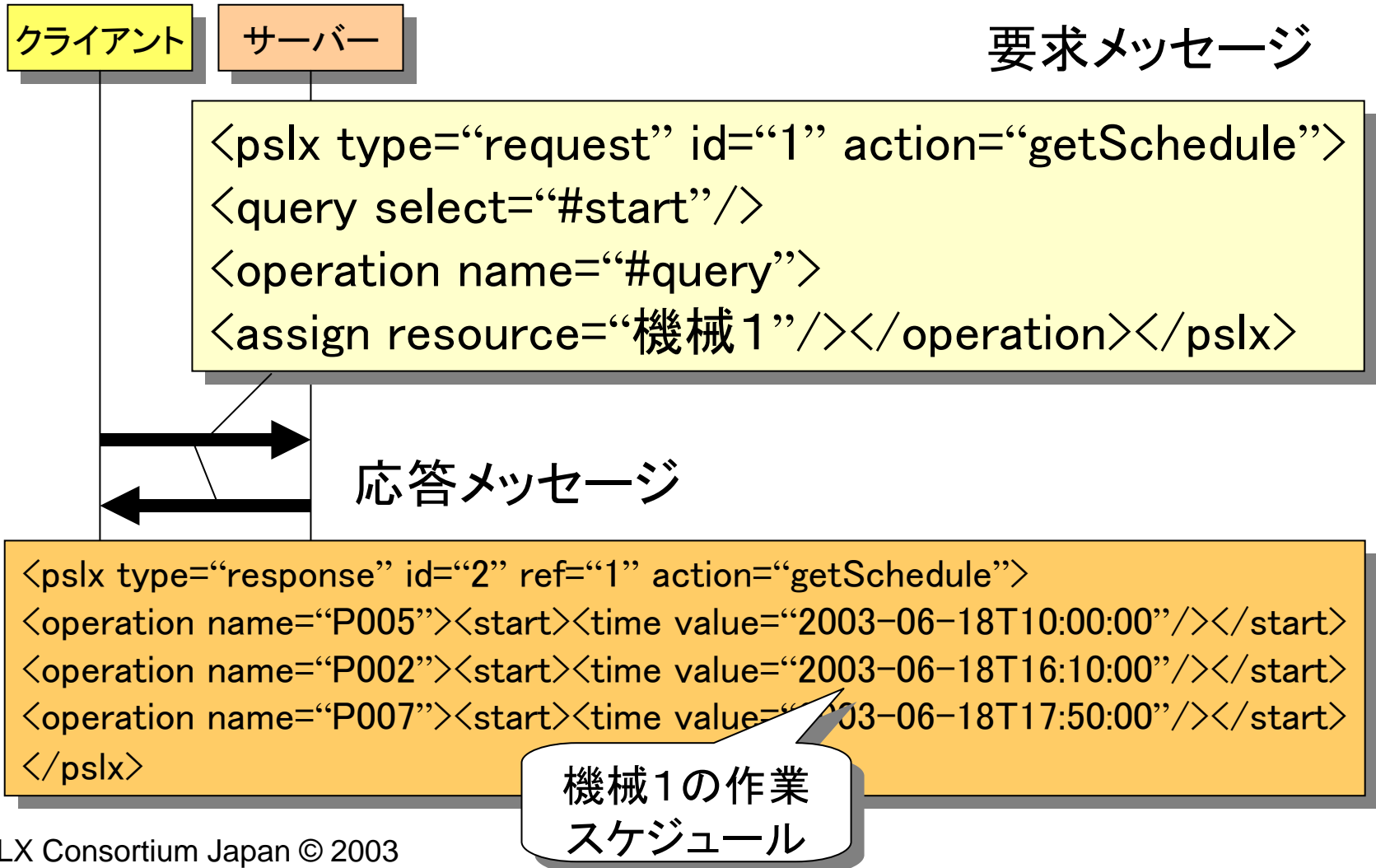




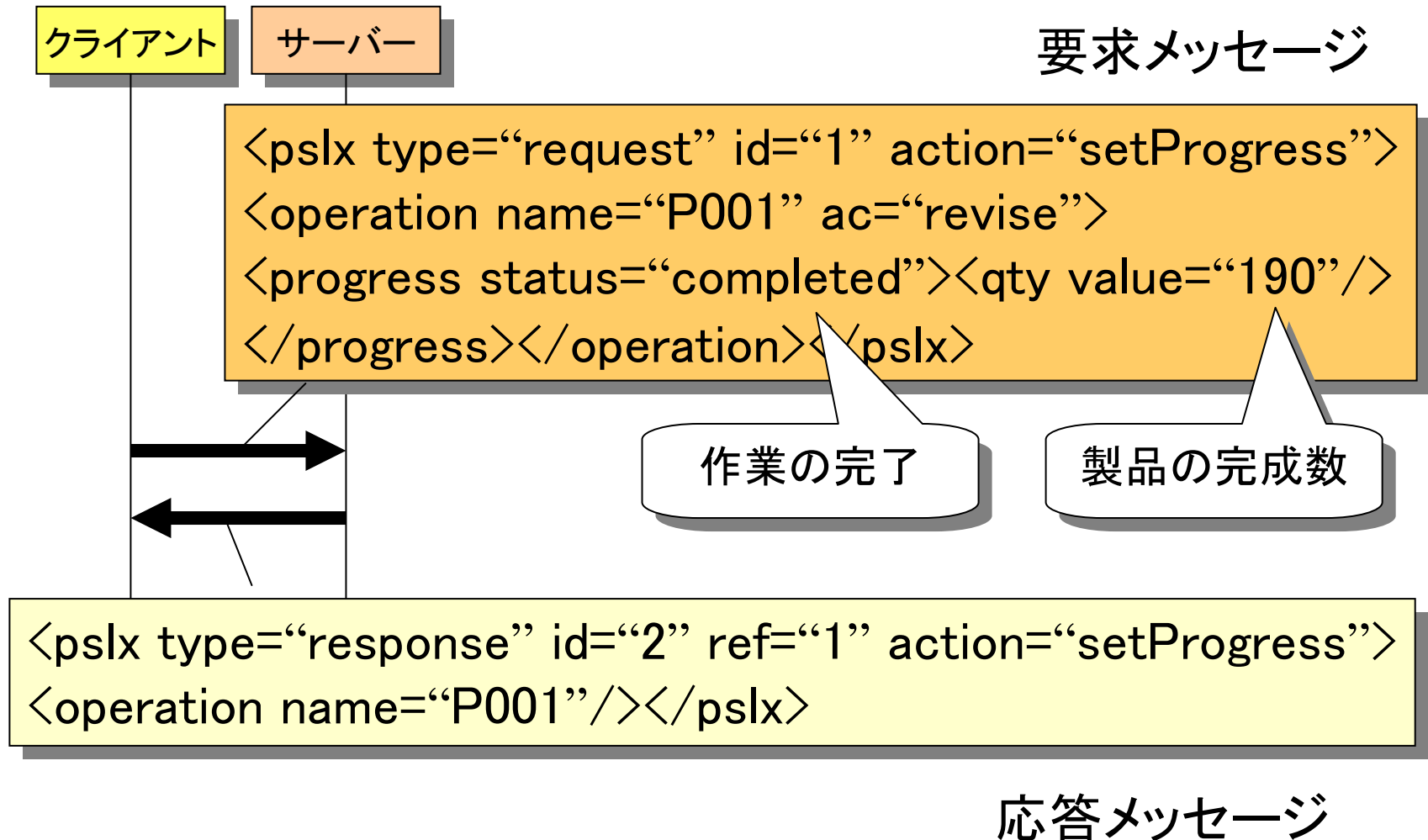
# 納期照会 (getOrder)



# スケジュール照会 (getSchedule)



# 進捗報告 (setProgress)

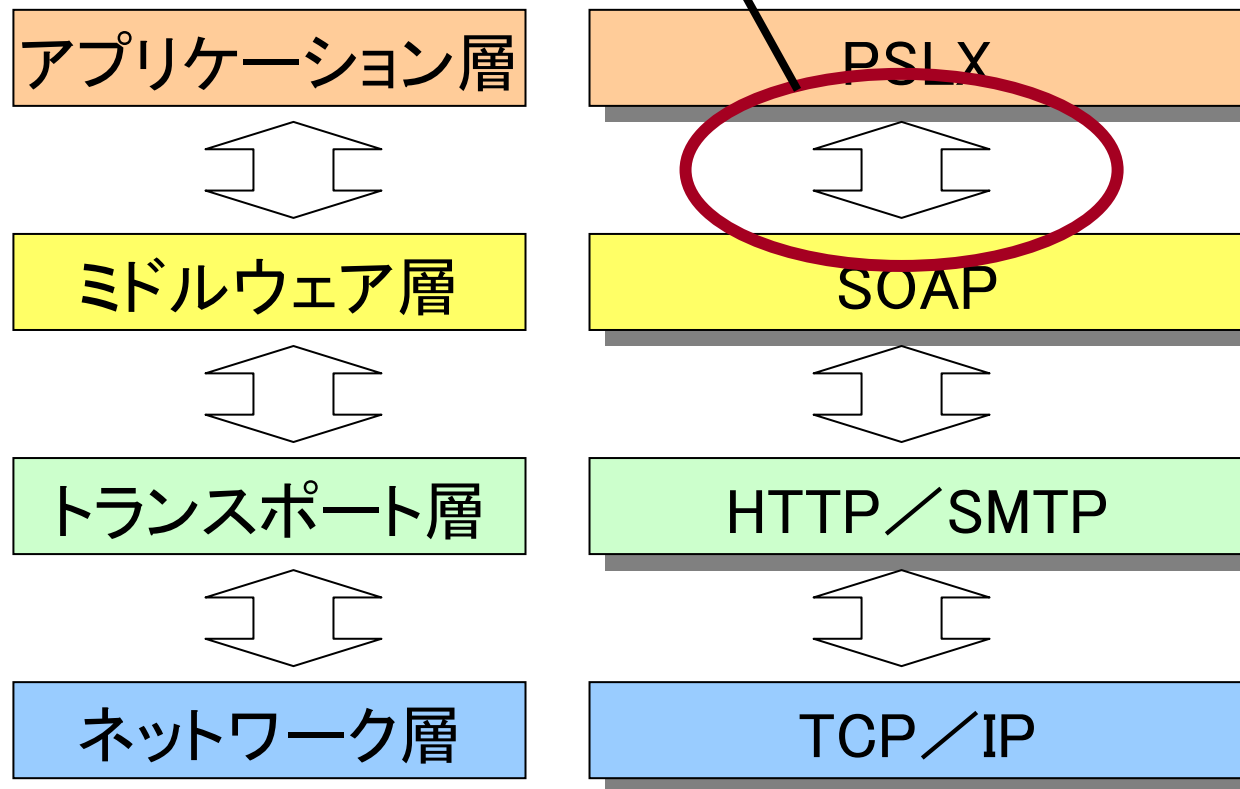


# もくじ

- はじめに
- パート1:XMLタグ仕様
- パート1:タグ表記サンプル
- パート1:照会／集計の表現
- パート2:PSLXメッセージ仕様
- パート2:メッセージ交換サンプル
- **パート3:PSLX通信規約**
- パート3:ミドルウェアとのバインド方法
- まとめ

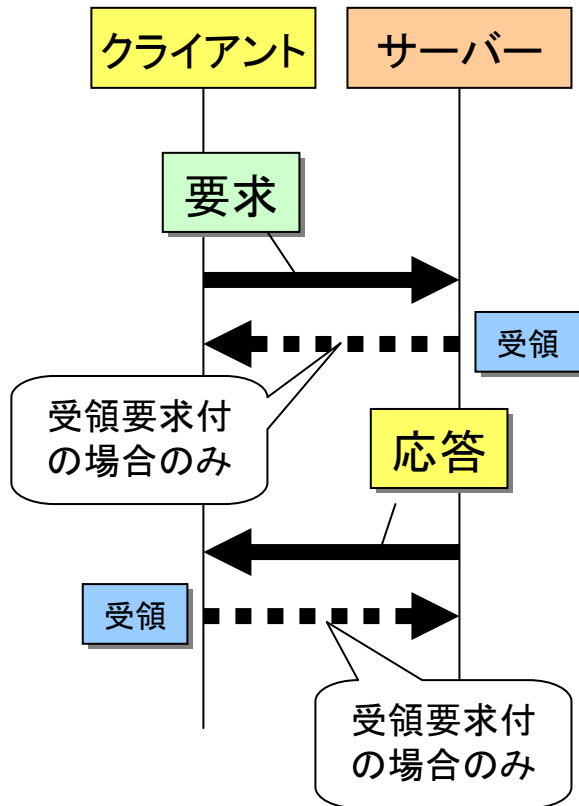
# ミドルウェアとの関係

本仕様書パート3のターゲット

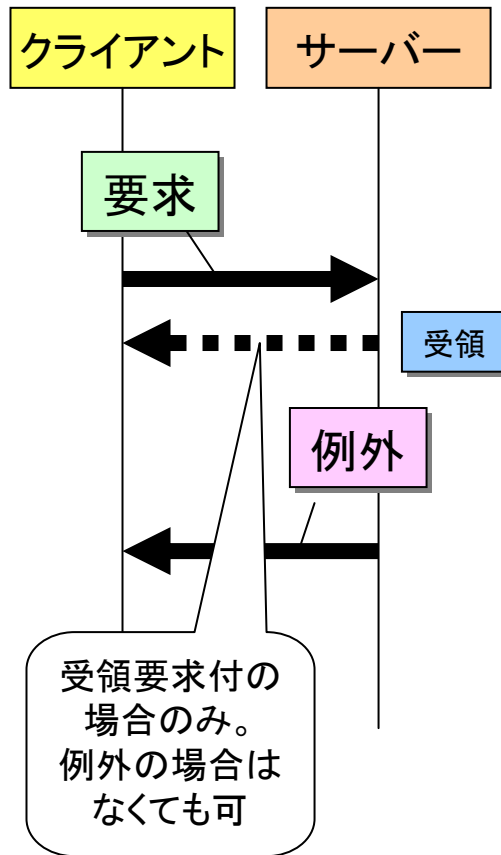


# メッセージ応答パターン

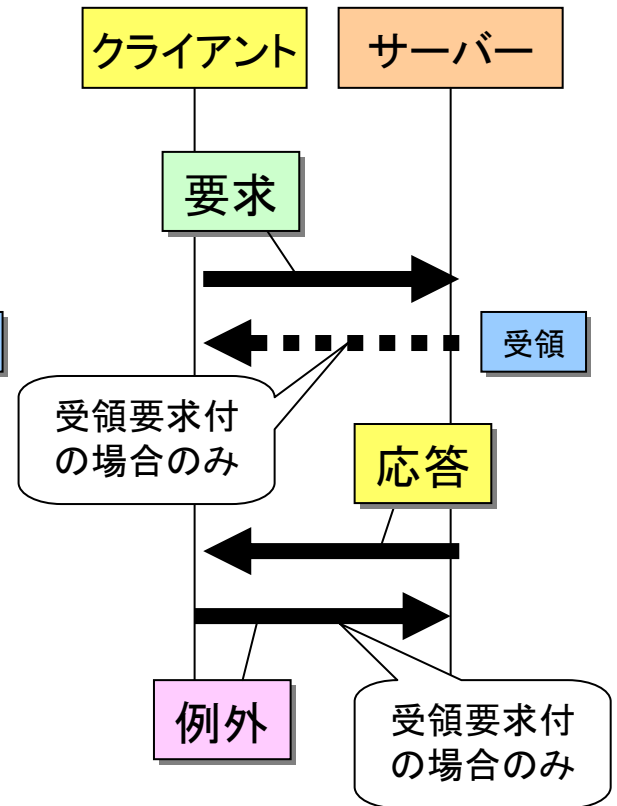
## 要求応答パターン



## 要求例外パターン



## 応答例外パターン



# HTTP(RPC型)へのバインディング

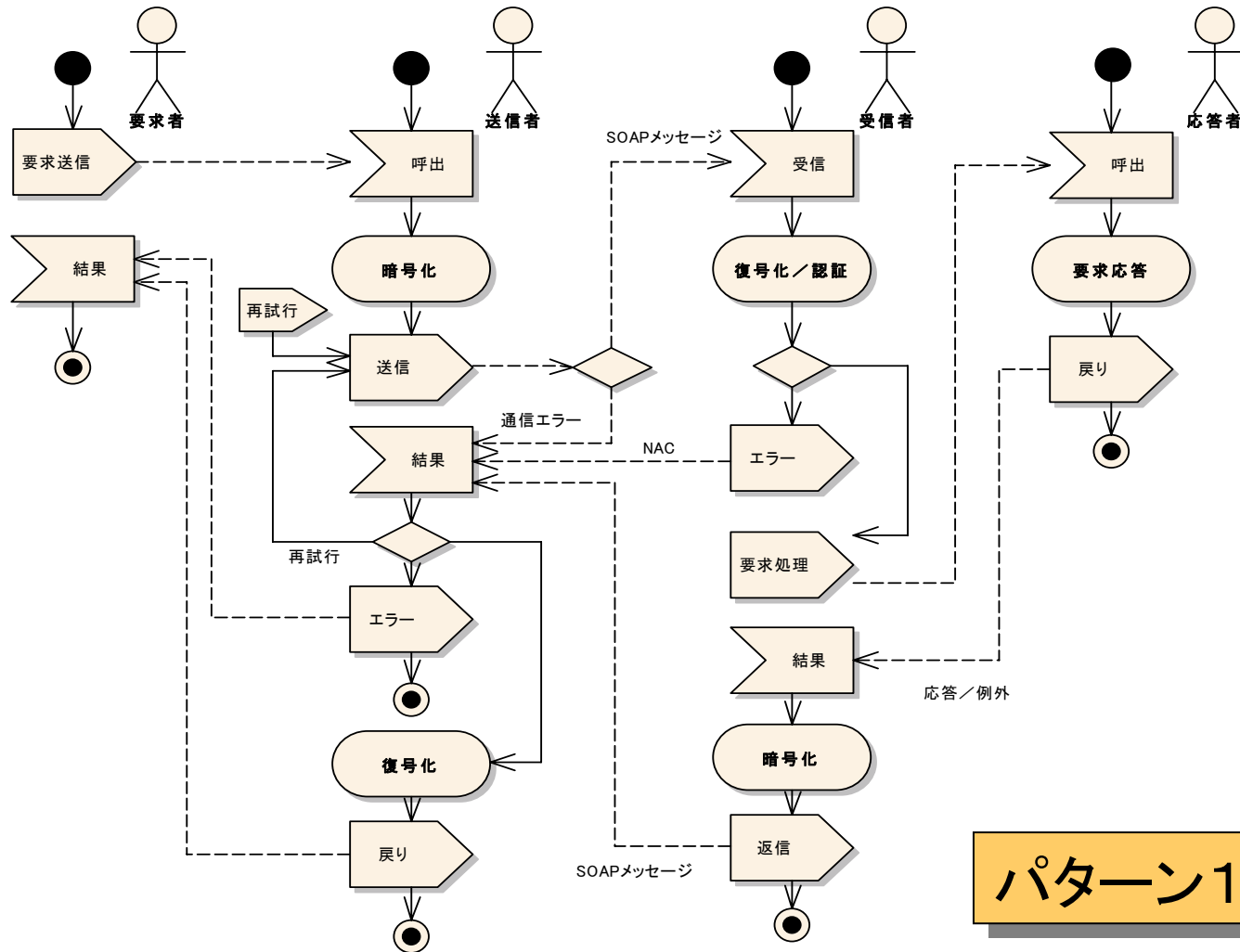
## 同期シーケンス用

パターンNO	往信	復信
1	要求メッセージ	応答メッセージ
2	要求メッセージ	例外メッセージ

## 非同期シーケンス用

3	要求メッセージ	なし
4	応答メッセージ	なし
5	受領メッセージ	なし
6	例外メッセージ	なし
7	なし	要求メッセージ
8	なし	応答メッセージ
9	なし	受領メッセージ
10	なし	例外メッセージ

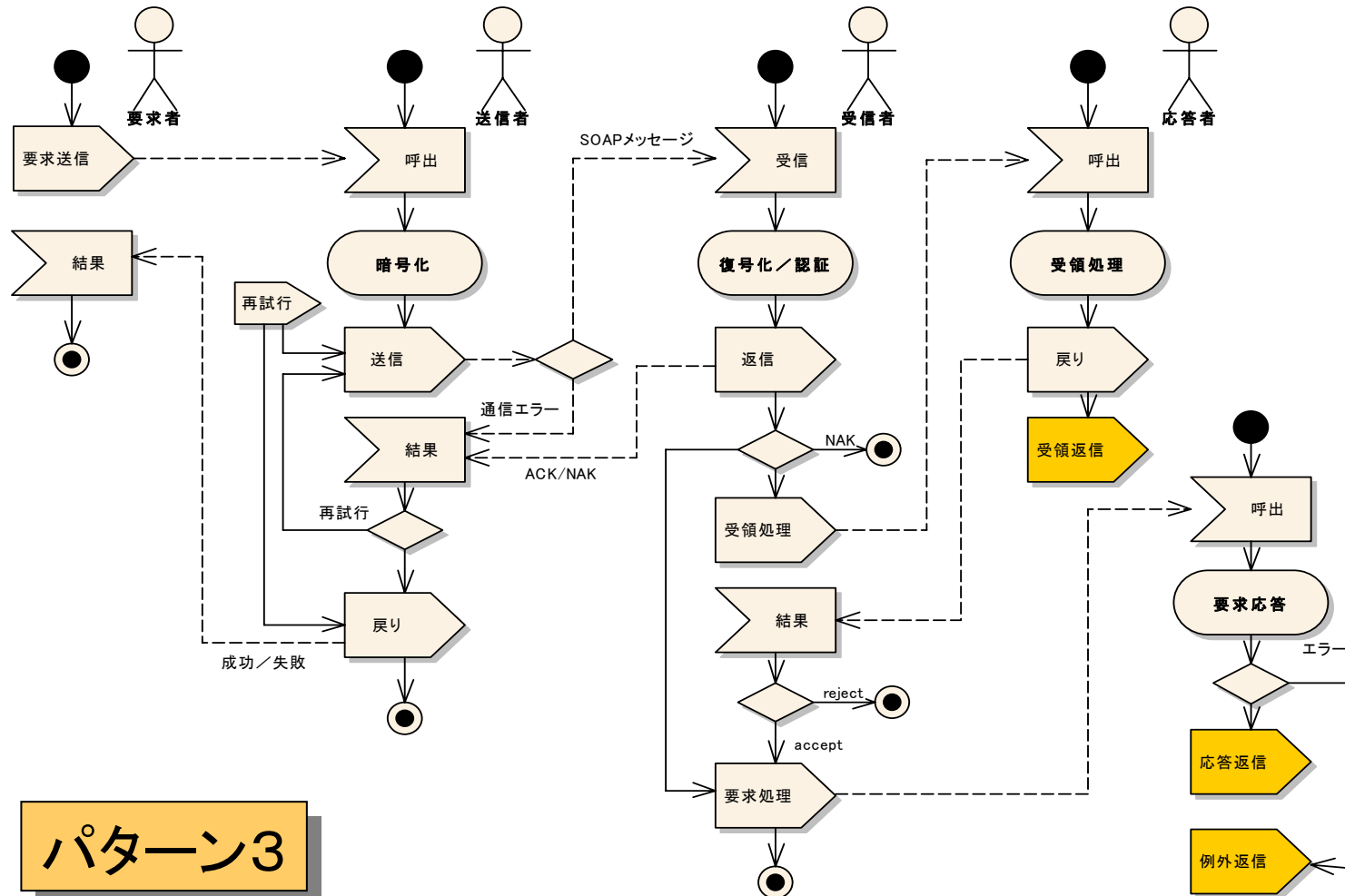
# 同期型シーケンスの例



パターン1, 2

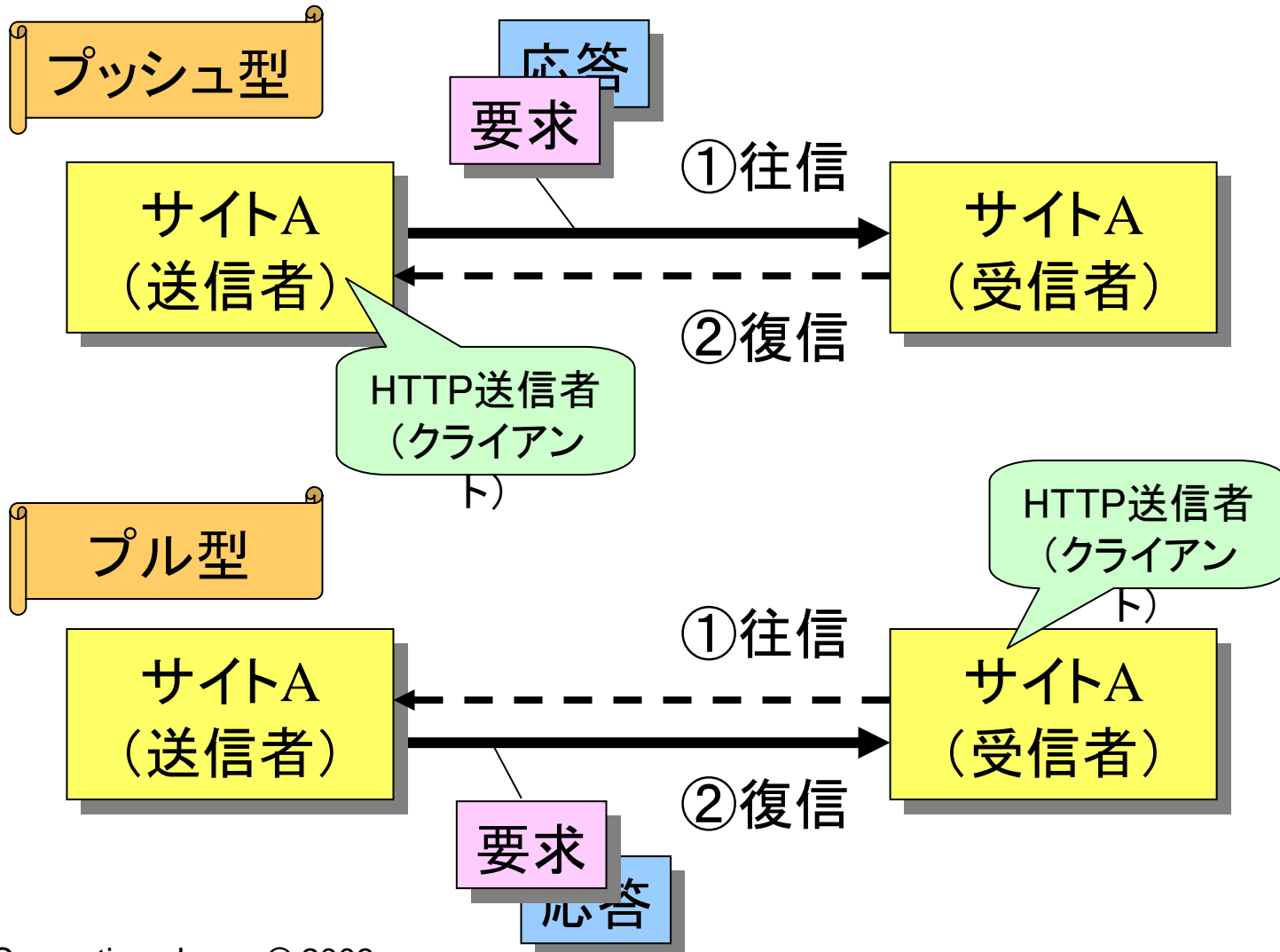


# 非同期型シーケンスの例（要求）



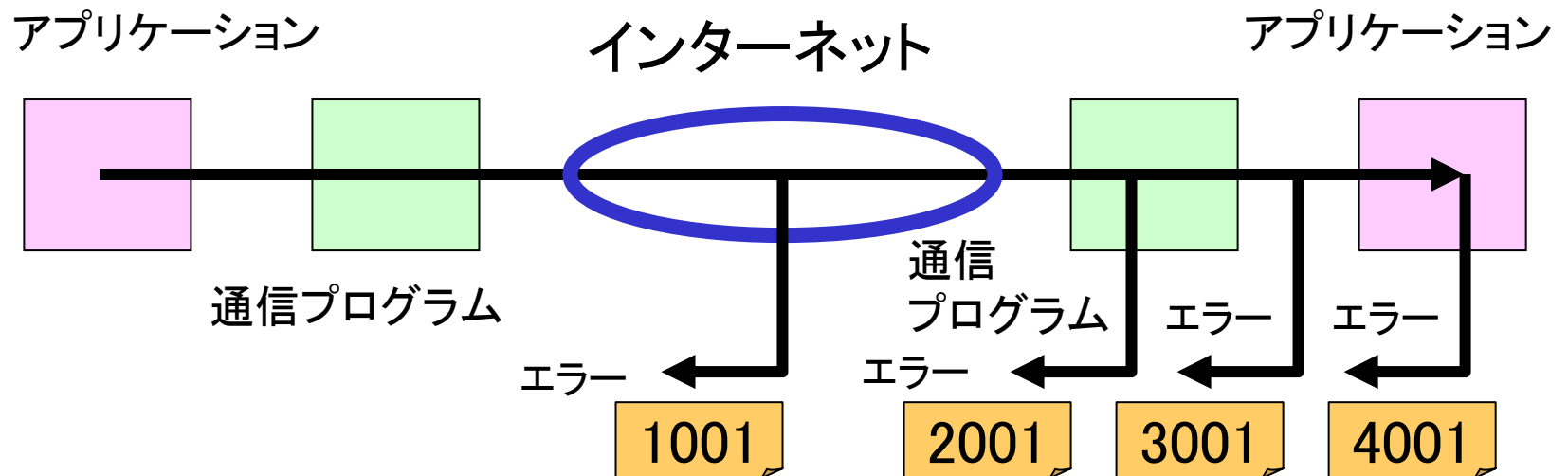
**パターン3**

# プル型への対応



# エラーの種類

コード	説明	発行元
1001～	ネットワーク上のエラー	トランスポート層
2001～	通信プログラム上でのエラー	ミドルウェア層
3001～	PSLXデータ解析エラー	PSLXインタフェース
4001～	アプリケーション上でのエラー	PSLXアプリケーション



# もくじ

- はじめに
- パート1:XMLタグ仕様
- パート1:タグ表記サンプル
- パート1:照会／集計の表現
- パート2:PSLXメッセージ仕様
- パート2:メッセージ交換サンプル
- パート3:PSLX通信規約
- **パート3:ミドルウェアとのバインド方法**
- まとめ

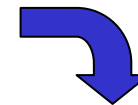
# ミドルウェアの選択

- Webサービス
  - メリット: 開発ツールが豊富、機構がシンプル
  - デメリット: セキュリティ機能が不足
- ebXMLメッセージサービス
  - メリット: グローバル組織、高信頼性通信
  - デメリット: 開発環境が不足、実績少ない
- RNIF (ロゼッタネット実装フレームワーク)
  - メリット: 運用実績が豊富、製品多数
  - デメリット: 特定業界、国際標準との関係

# Webサービスへのバインディング

```
<soapenv:Envelope>  
  <soapenv:Body>  
    <pslxMethodSync>  
      <pslxBody>...  
      PSLXメッセージ文字列...  
    </pslxBody>  
  <toLocal>プログラムID</toLocal>  
</pslxMethodSync>  
</soapevsn:Body>  
</soapenv:Envelope>
```

リクエスト



レスポンス

同期シーケンスの場合

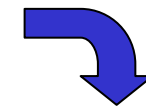
```
<soapenv:Envelope>  
  <soapenv:Body>  
    <pslxMethodSyncResponse>  
      <pslxMethodSyncReturn>  
      PSLXメッセージ文字列  
    </pslxMethodSyncReturn>  
  </pslxMethodSyncResponse>  
</soapenv:Body>  
</soapenv:Envelope>
```

# Webサービスへのバインディング

## 非同期シーケンス(プッシュ)の場合

```
<soapenv:Envelope>  
  <soapenv:Body>  
    <pslxMethodAsync>  
      <messageID>メッセージID</messageID>  
      <refID>参照メッセージID</refID>  
      <pslxBody>PSLXメッセージ文字列</pslxBody>  
      <toLocal>受信プログラムID</toLocal>  
      <fromURI>送信サーバーURI</fromURI>  
      <fromLocal>送信プログラムID</fromLocal>  
      <clientID>送信クライアントID</clientID>  
    </pslxMethodAsync>  
  </soapenv:Body>  
</soapenv:Envelope>
```

リクエスト



レスポンス

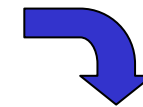
```
<soapenv:Envelope>  
  <soapenv:Body>  
    <pslxMethodAsyncResponse>  
      <pslxMethodAsyncReturn>  
        メッセージID  
      </pslxMethodAsyncReturn>  
    </pslxMethodAsyncResponse>  
  </soapenv:Body>  
</soapenv:Envelope>
```

# Webサービスへのバインディング

```
<soapenv:Envelope>  
  <soapenv:Body>  
    <pslxMethodPull>  
      <toLocal>受信プログラムID</toLocal>  
      <fromURI>送信サーバーURI</fromURI>  
      <fromLocal>送信プログラムID</fromLocal>  
    </pslxMethodPull>  
  </soapenv:Body>  
</soapenv:Envelope>
```

非同期シーケンス(プル)の場合

リクエスト



レスポンス

```
<messageID>メッセージID</messageID>  
<refID>参照メッセージID</refID>  
<pslxBody>PSLXメッセージ文字列</pslxBody>  
<fromURI>送信サーバーURI</fromURI>  
<fromLocal>送信プログラムID</fromLocal>  
<clientID>送信クライアントID</clientID>
```

```
<soapenv:Envelope>  
  <soapenv:Body>  
    <pslxMethodPullResponse>  
      <pslxMethodPullReturn>  
        メッセージ内容  
      </pslxMethodPullReturn>  
    </pslxMethodPullResponse>  
  </soapenv:Body>  
</soapenv:Envelope>
```

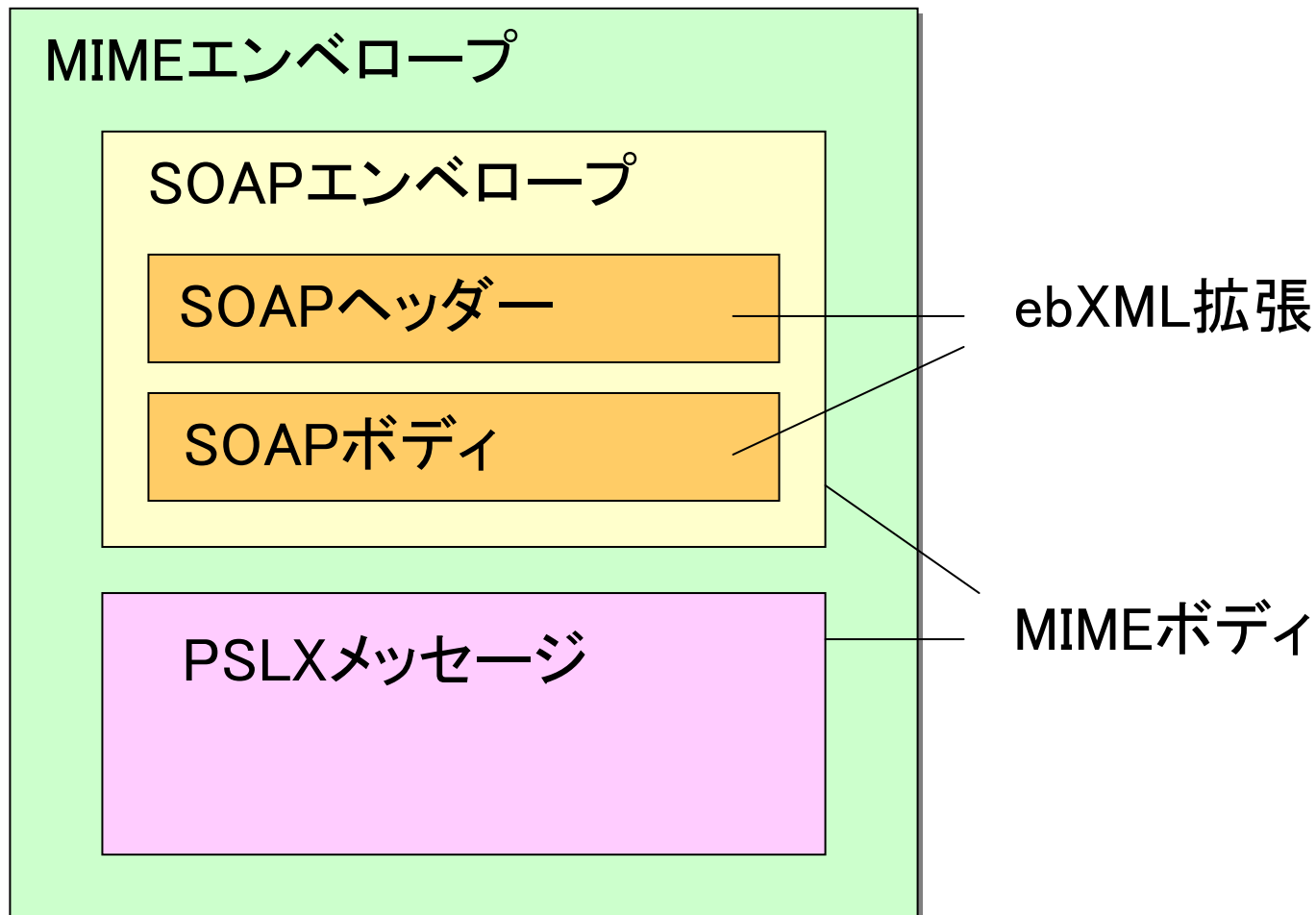


# Webサービスへのバインディング

タグ名	データタイプ	同期	非同期要求	非同期応答
toLocal	文字列	△	△	△
fromURI	文字列	×	○	×
fromLocal	文字列	×	△	×
clientID	文字列	×	○	○
messageID	文字列	×	○	○
refID	文字列	×	×	○
pslxBody	文字列	○	○	○

○・・・必須  
△・・・オプションナル  
×・・・不可

# ebXML-MSへのバインディング



# ebXML-MSへのバインディング

```
<soapenv:Envelope>
  <soapenv:Body>
    <MessageHeader>
      <From>
        <PartyID type="URI">送信サーバーURI</PartyID>
        <PartyID type="local">送信プログラムID</PartyID>
        <PartyID type="client">送信クライアントID</PartyID>
      </From>
      <To>
        <PartyID type="URI">受信サーバーURI</PartyID>
        <PartyID type="local">受信プログラムID</PartyID>
        <PartyID type="client">受信クライアントID</PartyID>
      </To>
      <Service>PslxServise</Service>
      <Action>インタフェース名</Action>
      <MessageData>
        <MessageId>メッセージID</MessageId>
        <RefToMessageId>参照メッセージID</RefToMessageId>
      </MessageData>
    </MessageHeader>
  </soapenv:Body>
</soapenv:Envelope>
```

} 送信元  
情報

} 送信先  
情報

} インタフェース名

} メッセージ  
識別情報

# ebXML-MSへのバインディング

タグ名	データタイプ	要求	応答
From:PartyID:URI	文字列	○	×
From:PartyID:local	文字列	△	×
From:PartyID:client	文字列	○	×
To:PartyID:URI	文字列	○	○
To:PartyID:local	文字列	△	△
To:PartyID:client	文字列	×	○
Service	文字列	○	○
Action	文字列	○	○
MessageId	文字列	○	○
RefToMessageId	文字列	×	○

# もくじ

- はじめに
- パート1:XMLタグ仕様
- パート1:タグ表記サンプル
- パート1:照会／集計の表現
- パート2:PSLXメッセージ仕様
- パート2:メッセージ交換サンプル
- パート3:PSLX通信規約
- パート3:ミドルウェアとのバインド方法
- まとめ

# システム実装のレベル

- レベル1
  - とりあえず、最低限の機能を実装
- レベル2
  - 一般的に必要な十分な機能を実装
- レベル3
  - PSLX仕様のフル機能を実装

# どこから手をつかるか？

- ステップ1
  - 永続化機能(PSLXファイルの読み書き)をレベル1またはレベル2で実装する
- ステップ2
  - 特定のインタフェース(getOrderなど)を2、3選  
び、レベル1で実装する
- ステップ3
  - 対応するインタフェースの種類や実装レベル  
を必要に応じて増やす

## まとめ

- PSLXにおけるXML利用の方針と位置付けを説明
- XMLを利用したタグによる問題記述方法の説明
- 業務プロセスにおけるメッセージ交換方法の説明
- ミドルウェアプログラムを用いた通信方式の説明