SQL Server 2005
Visual Studio 2005
スキルアップセミナー

【Advanced トラック:9】
SQL Server 2005:
分散データベースアプリケーション開発

会社名：有限会社アークウェイ
役職：コンサルタント
名前：黒石　高広

Microsoft

自己紹介

- ECサイト構築、EAIシステム構築のコンサルティングを経て、現在.NETでのアジャイル開発のコンサルティングに携わる
- 有限会社アークウェイ
  http://www.archway.co.jp/
- NAgile
  - .NET技術でのアジャイル開発スタイル

Agenda

- Demo
- XML Webサービス
- Service Broker
- Notification Service
- アプリケーション統合のヒント
- まとめ

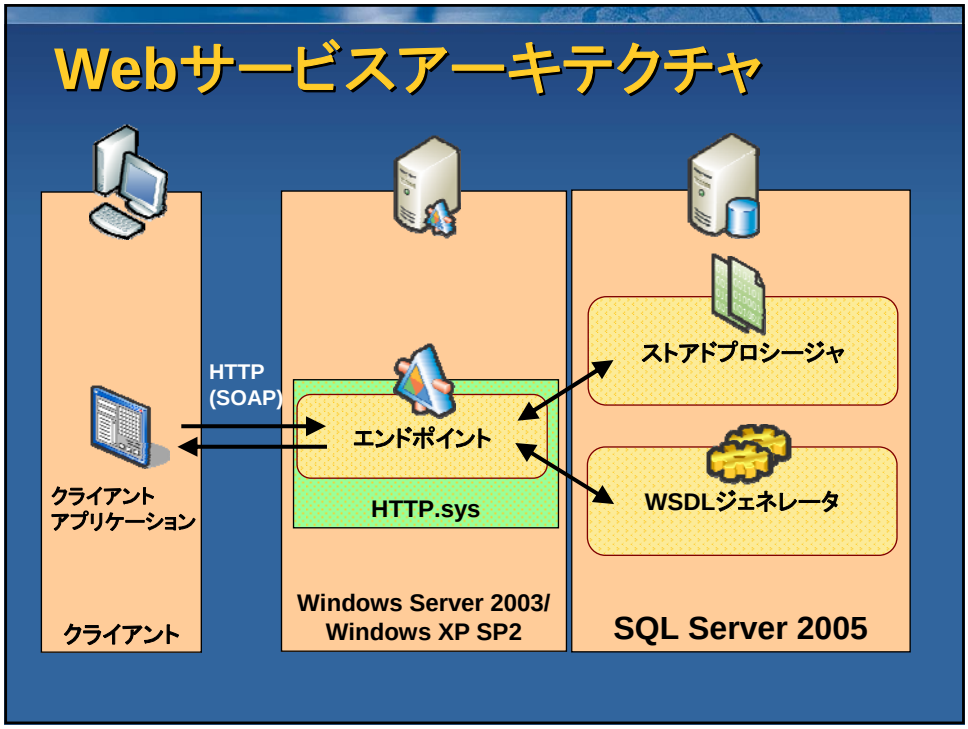
目的

- 分散環境にあるデータベースをSQL Server 2005の新機能を使ってどのように統合できるか学ぶ
- アプリケーション統合の際に、役立つ様々なMicrosoft技術が、どのような状況／局面にどの技術が最適か学ぶ

XML Webサービス概要

- SQL Serverのストアドプロシージャをwebサービスとして公開し、クライアントから直接データベース操作を行えるようにする機能
- SQL Server 2000ではSQLXML機能として利用可能
  - 最新版：SQLXML 3.0 sp2
  - 下記URLよりダウンロード可能

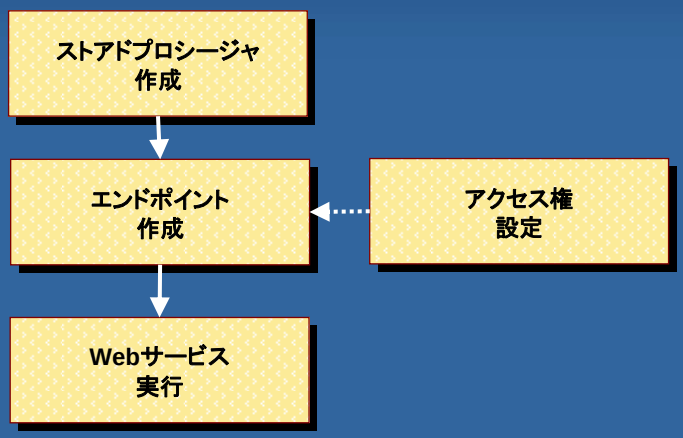
    http://www.microsoft.com/downloads/details.aspx?displaylang=ja&FamilyID=4C8033A9-CF10-4E22-8004-477098A407AC



Webサービスアーキテクチャ

- Internet Information Serviceを利用せず、直接カーネルモードの（HTTP.sys）を利用
  - Windows Server 2003 / Windows XP sp2
  - SQL Server 2005上でWebサービスを実行
- クライアントから構成されたエンドポイントにアクセスすると、WSDLの参照やストアドプロシージャの実行が可能

Webサービス構成の流れ



Webサービス構成の流れ

1. ストアドプロシージャの作成
   - Webサービスとして公開するストアドプロシージャの作成
2. エンドポイントの作成
   - HTTPやSOAPに関する設定を行い、実行するストアドプロシージャと関連付ける
3. アクセス権の設定(Option)
   - エンドポイントに対するアクセス権の設定
4. Webサービス実行
   - クライアントからWSDLを参照し、Webサービスを実行

ストアドプロシージャの作成

```
CREATE PROCEDURE sp_SelectEmployees
    ( @emp_id int )
AS BEGIN
    SELECT
      [EmployeeID], [LastName], [FirstName], [Title]
    FROM
      [Northwind].[dbo].[Employees]
    WHERE
      EmployeeID = @emp_id
END
```

ストアドプロシージャの作成

- 作成するストアドプロシージャに対する制約は特に無し
- 注意事項
  - Webサービスで利用できないストアドプロシージャの引数名
    - “@”, “@@”, “@@param_name”

エンドポイントの作成

```
CREATE ENDPOINT HelloWorldEndPoint
      STATE = STARTED
AS HTTP (
      AUTHENTICATION = ( INTEGRATED ),
      PATH = '/sql/demo',
      PORTS = ( CLEAR )
)
FOR SOAP (
      WEBMETHOD (
          'http://www.archway.co.jp/sqlsample/', 'HelloWorld'
          ( NAME = 'Archway.dbo.sp_HelloWorld' ),
          WSDL = DEFAULT
      )
)
```

Webサービスの認証モード

HTTPアクセスするURLのパス名

Webメソッド名

このWebメソッドで実行するストアドプロシージャ

エンドポイントの作成

- エンドポイントの設定
  - エンドポイント名の指定
  - STATEの設定
    - Webサービスの状態、通常はSTARTEDで構成
- HTTPの設定
  - 認証モード
  - パス
    - URLのパス名の設定
- SOAPの設定
  - Webメソッドの設定
    - 名前空間、Webメソッド名の設定
    - 実行するストアドプロシージャの設定
    - WSDLの設定

Webサービスの認証モード

- IISの認証モードとほぼ同等
  - BASIC
    - 基本認証、インターネット経由での認証で利用
    - ユーザ名／パスワードが生データのまま送信
  - DIGEST
    - ダイジェスト認証、基本認証を拡張しユーザ名／パスワードに1方向ハッシュ化
  - KERBEROS
    - 秘密鍵暗号を利用した認証方式、インターネット標準
    - Windows 2000以降でサポート
  - NTLM
    - Windows 9X, Windows NT 4.0などの旧クライアントで利用される認証モード
  - INTEGRATED
    - 統合認証、アクセスするWindowsユーザの資格でストアドプロシージャを実行
- 匿名認証はサポートされていないので注意

アクセス権設定(Option)

- エンドポイントへのアクセス権設定
  - CONNECT ON ENDPOINT
    - エンドポイントへのHTTP接続権限、WSDL参照の権限設定
- エンドポイントの管理権限設定
  - ALTER ANY ENDPOINT
  - ALTER ON ENDPOINT
  - CONTROL ON ENDPOINT

※ Beta版から構文の変更があるため、最新情報はBooksOnlineを参照

Webサービス実行

- クライアントからの呼び出し例

```
localhost.proxy soapClient = new localhost.proxy();
soapClient.Credentials =
    System.Net.CredentialCache.DefaultCredentials;
soapClient.HelloWorld("Hello World!!");
```

- .NETでの実装は、通常のASP.NET Webサービスの呼び出しとほぼ同じ
  - 統合認証を利用する場合は、SOAPクライアントのCredentialCacheを書き換える

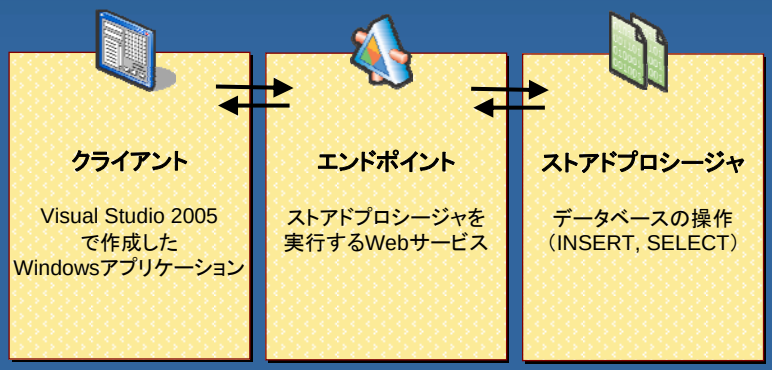
その他

- *SELECT * FROM sys.http_endpoints*
  - 登録されているエンドポイントの一覧表示
- DROP ENDPOINT *<エンドポイント名>*
  - エンドポイントの削除

demo

Webサービス構築

デモの構成



ASP.NET Webサービスとの比較

- 単純な処理を行った際の処理速度
  - 階層が少ない分、早い処理速度が期待できるため
- 構成の容易さ
  - 開発者の経験や感覚に依存するため
- 複雑な処理の開発効率
  - T-SQLよりも.NETのほうが豊富なAPIが提供されるため
- 負荷分散
  - SQL Server 2005上でのみWebサービスを実行できるため負荷分散（スケールアウト）は難しい

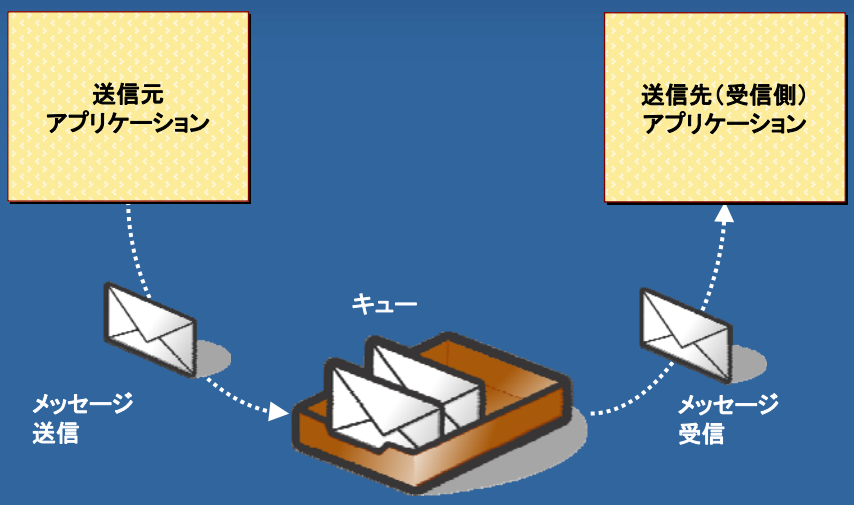
Service Broker概要

- SQL Server 2005間でメッセージを送受信し、アプリケーション間のデータ統合を行うための機能
  - 非同期連携
  - 順序制御
- ストアドプロシージャから外部サービスへメッセージ送信
  - 別のストアドプロシージャを非同期で実行
- SQL Server 2005での新機能

Service Broker適用シナリオ

- 非常に多くの処理を行っている巨大なストアドプロシージャをService Brokerの機能を使っていくつかに分割
  - 複数のストアドプロシージャを並列実行
  - ただし、例外処理が複雑になるので注意
- Webサービス機能で受信した巨大なサイズのメッセージをService Brokerの機能を使って非同期処理化
- 外部データベースへ送るデータを、Service Brokerの機能を使って順序制御

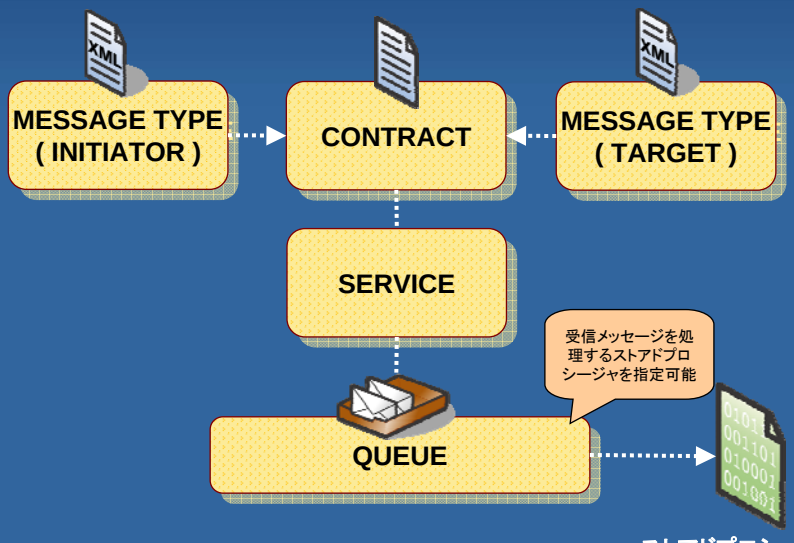
キュー機能



キュー機能

- 非同期連携
  - 送信元アプリケーションからキューにメッセージを送信した時点で別の処理を続行可能
    - 応答メッセージを待つ必要は無い
  - 同期連携と比較すると例外処理が難しい
- 順序制御
  - キューにメッセージを送信した順番で、メッセージが受信・処理される（First In Fisrt Out）

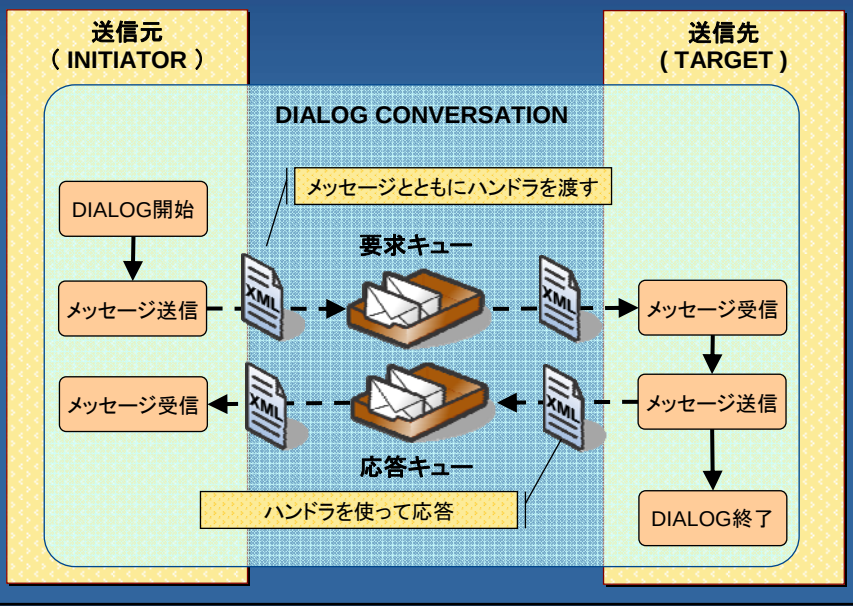
Service Broker論理構成



Service Broker論理構成

- MESSAGE TYPE
  - メッセージ検証の方式を定義
- CONTRACT
  - メッセージ送受信の取り決め、送信元のメッセージタイプや送信先のメッセージタイプを定義
- QUEUE
  - メッセージを蓄えるキュー
  - キュー内のメッセージを処理するストアドプロシージャを定義可能
- SERVICE
  - QUEUEとCONTRACTの関連付け
  - サービス名は送信時に使用

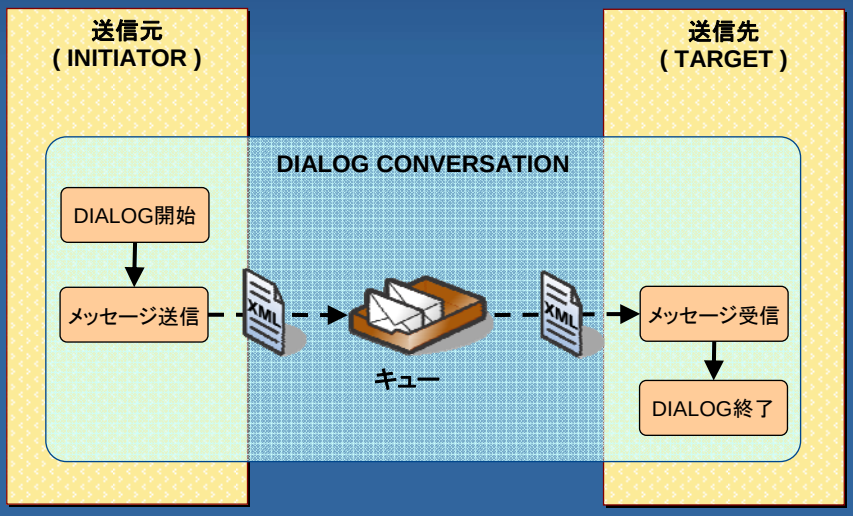
メッセージ送受信

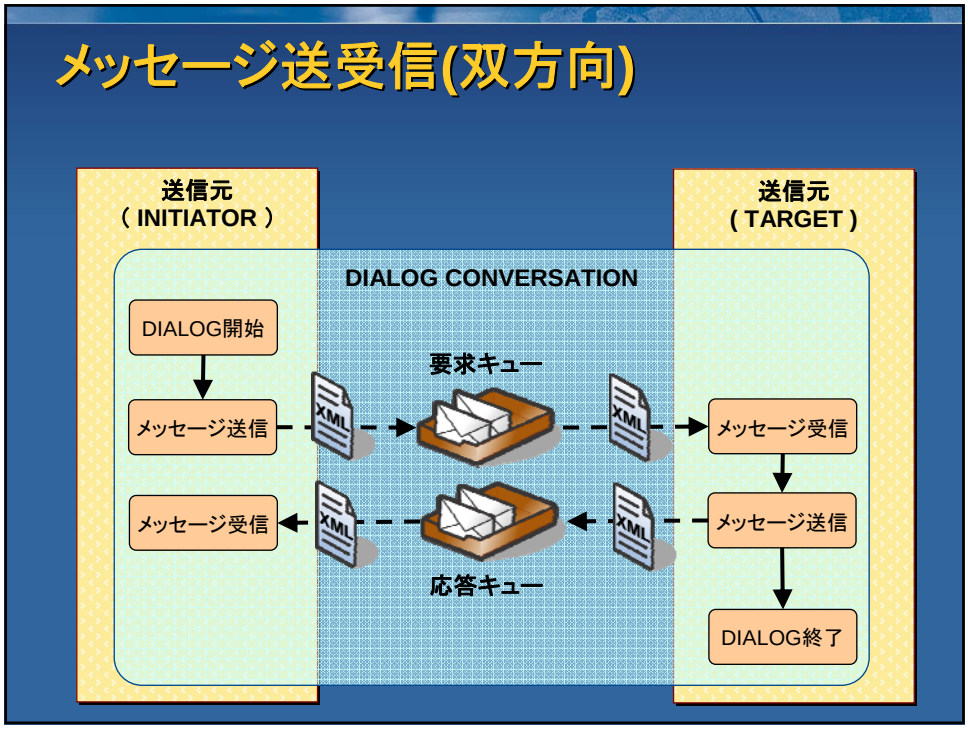


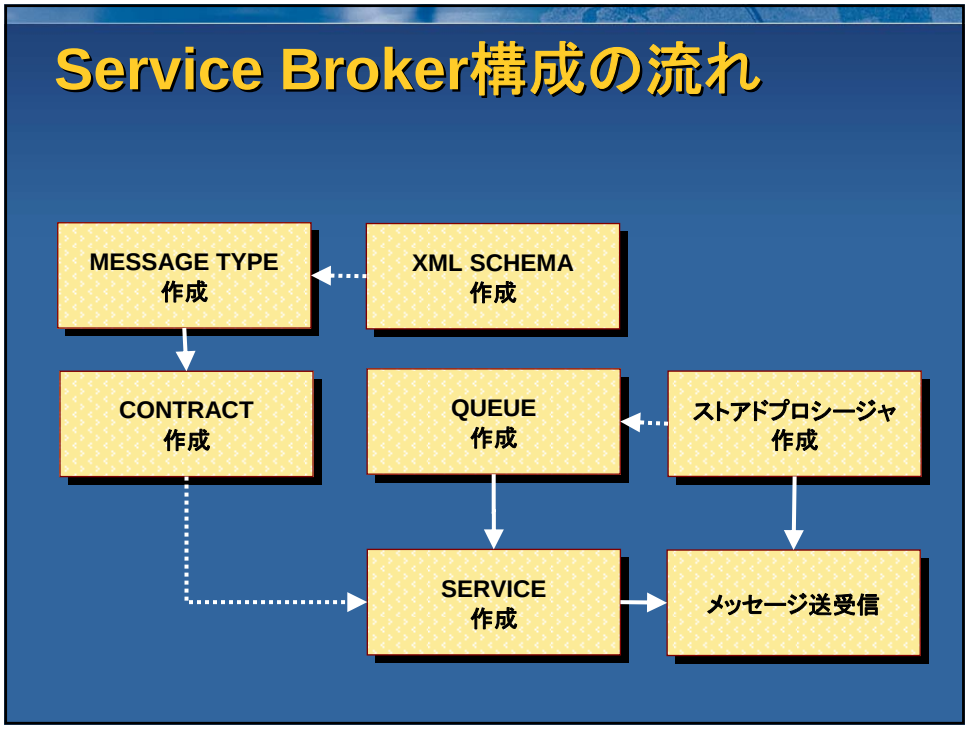
メッセージ送受信

- DIALOG CONVERSATIONを利用してメッセージ送受信
  - トランザクション処理に近いイメージ
- メッセージ送信、受信によって役割が異なる
  - INITIATOR
    - メッセージ送信側のサービス
  - TARGET
    - メッセージ受信側のサービス
- Conversation Handlerを利用して、双方向のメッセージ送受信

メッセージ送受信(1方向)







Service Broker構成の流れ

- XML SCHEMAの作成(Option)
  - XMLスキーマ(XSD)によるメッセージの検証を行いたい場合に必要
- MESSAGE TYPEの作成
  - メッセージ検証の方式を設定
- CONTRACTの作成
  - メッセージ送受信の契約(Contract)を設定
- QUEUEの作成
  - メッセージを蓄えるキューの作成
- SERVICEの作成
  - QUEUEとCONTRACTを関連付けるサービスの作成
- ストアドプロシージャの作成
  - メッセージ受信用ストアドプロシージャの作成
  - メッセージ送信用ストアドプロシージャの作成

XML SCHEMAの作成(Option)

```
CREATE XML SCHEMA COLLECTION
    HelloWorldXml
AS
    N'<?xml version="1.0"? encoding="UTF-16">
    <xsd:schema targetNamespase="http://">
    ...
    </xsd:schema>';
```

XSDの文字列

- XMLスキーマ(XSD)による厳密なメッセージの検証を行いたい場合に必要な作業

MESSAGE TYPEの作成

```
CREATE MESSAGE TYPE
    [//archway.co.jp/sample/HelloWorld]
    VALIDATION = WELL_FORMED_XML ;
```

- メッセージ検証の種類を指定
  - NONE
    - メッセージの検証を行わない
  - EMPTY
    - 必ず空のメッセージ(Null)であるか検証
  - WELL_FORMED_XML
    - XMLのメッセージであるか検証
  - VALID_XML WITH COLLECTION
    - XMLスキーマによるメッセージの検証

CONTRACTの作成

```
CREATE CONTRACT HelloWorldContract
(
    SENT BY INITIATOR HelloWorldRequest,
    SENT BY TARGET HelloWorldResponse
);
```

TARGET側のメッセージタイプ

TARGET側のメッセージタイプ

CONTRACTの作成

- MESSAGE TYPEとメッセージ送受信の方向を定義
  - SENT BY INITIATOR
    - INITIATOR側から送信されるメッセージタイプの指定
  - SENT BY TARGET
    - TARGET側から送信されるメッセージタイプの指定
  - SENT BY ANY
    - INITIATOR , TARGETどちら側からでも送信可能なメッセージタイプ
- 既定では、スキーマ検証を行わない設定
  - [DEFAULT] SENT BY ANY
- 双方向のメッセージ送受信を行い、かつスキーマの検証を行う場合はINITIATOR, TARGETを指定

QUEUEの作成

```
CREATE QUEUE HelloWorldQueue
    WITH
      ACTIVATION (
          PROCEDURE_NAME = sp_HelloWorldReceive,
          MAX_QUEUE_READERS = 1,
          EXECUTE AS SELF
      );
```

- メッセージ送受信で利用するキューの作成
- 受信したメッセージを処理するストアドプロシージャを指定可能(Option)

SERVICEの作成

```
CREATE SERVICE HelloWorldService
    ON QUEUE HelloWorldQueue(
      HelloWorldContract
    )
```

- QUEUEとCONTRACTの関連付け
- CONTRACTを指定しなかった場合は、INITIATOR専用のサービスとなる

メッセージ送信用ストアドプロシージャ

```
SET NOCOUNT ON
DECLARE @conversationHandler uniqueidentifier;

BEGIN TRANSACTION

BEGIN DIALOG @conversationHandler
    FROM SERVICE
    TO SERVICE
    ON CONTRACT
    WITH Encryption = OFF, Lifetime = 600;

SEND ON CONVERSATION @conversationHandler
    MESSAGE TYPE HelloWorldMessage( N'Hello World!!' )

COMMIT
```

DIALOGの開始

メッセージ送信

メッセージ受信用ストアドプロシージャ

```
SET NOCOUNT ON
DECLARE @conversationHandle uniqueidentifier
DECLARE @message_body nvarchar(MAX);

BEGIN TRANSACTION;

RECEIVE TOP(1)
      @conversationHandle = conversation_handle,
      @message_body = message_body
FROM HelloWorldTargetQueue

SEND ON CONVERSATION @conversationHandle
      MESSAGE TYPE HelloWorldResponse
         (N'Hello From ' + @@servername)

END CONVERSATION @conversationHandle

COMMIT
```

メッセージ受信

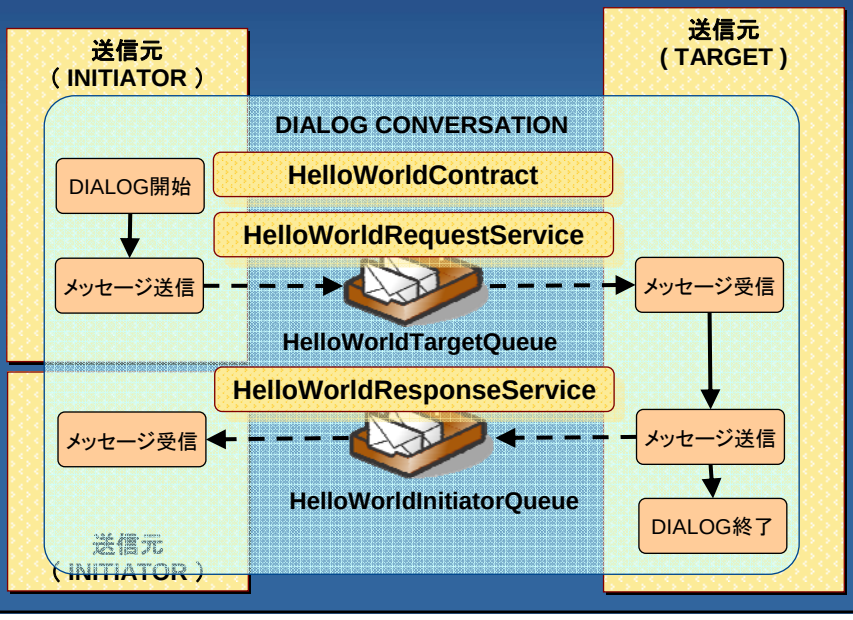
応答メッセージ送信

DIALOG終了

demo

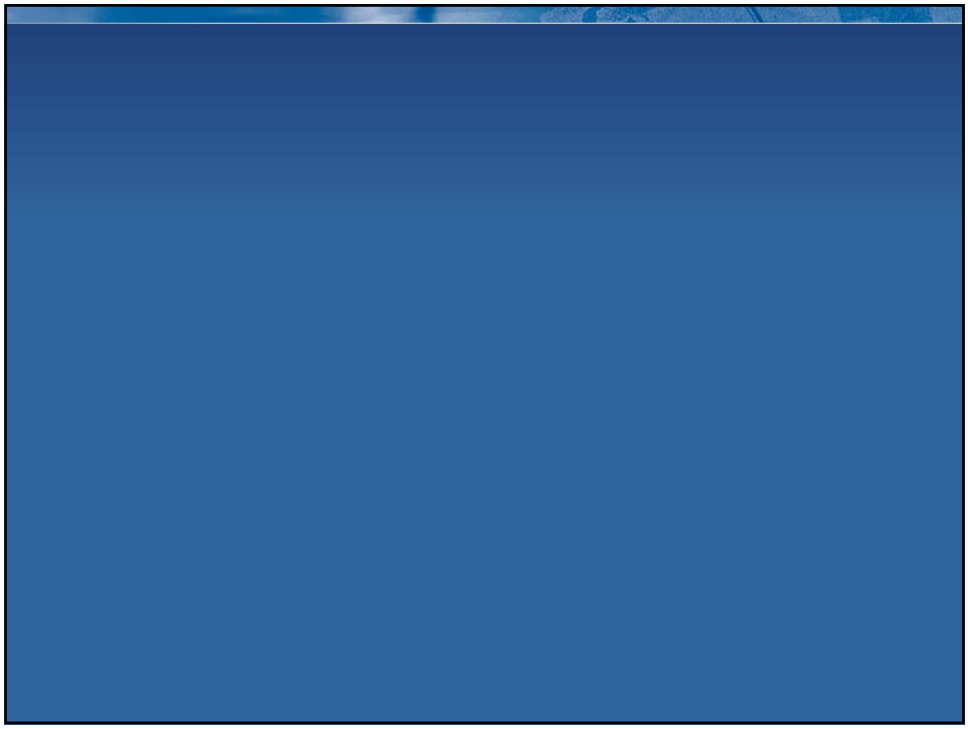
Service Brokerを利用したメッセージ交換

デモの構成



MSMQとの比較

- メッセージサイズの制限
  - MSMQと違って、1メッセージのサイズが4MBまでという制限が無い
- SQL Server 2005同士でないと連携できない
  - ただし、Integration Servicesなどを利用し、旧バージョンや他製品／外部アプリケーションとの連携可能
- 開発効率
  - MESSAGE TYPE, QUEUE作成のGUIが無い



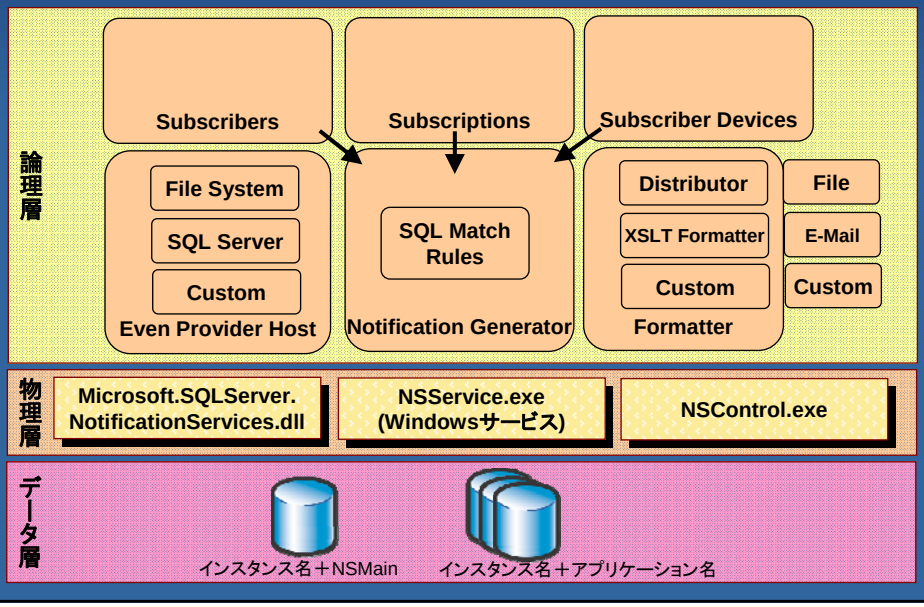
Notification Service概要

- ファイルシステムやデータベースのデータ挿入、変更などのイベントを拾い、ユーザに対してメールやファイル出力などで通知する機能
- SQL Server 2000でもNotification Serviceは利用可能
  - 下記URLからダウンロードし、追加インストール
    - http://www.microsoft.com/japan/sql/ns/

Notification Service適用シナリオ

- ユーザーへの通知
  - 株価が変動した場合、メールで通知するなどアプリケーションの通知機能として利用
- 管理者への通知
  - メールでの障害通知、RSSでのログ情報通知など管理タスクの補助機能として利用
- Custom Delivery Protocolを作成し、Windows Messanger(IM)とも連携可能

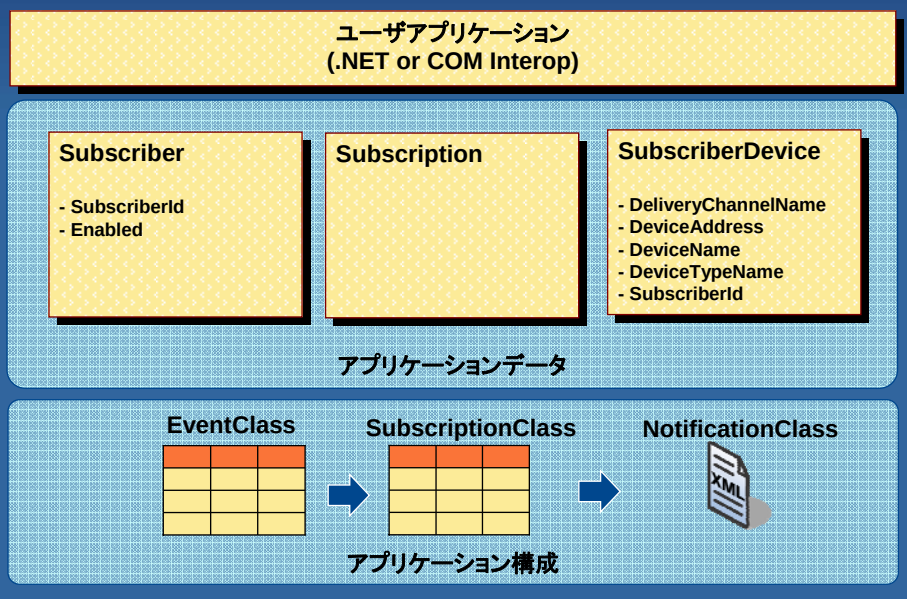
Notification Serviceアーキテクチャ



Notification Serviceアーキテクチャ

- 論理層
  - どのようにイベントを検知し、通知の生成や配信を行うか構成ファイルを使って定義
  - 構成情報はデータベースに展開
- 物理層
  - 論理層で定義された構成に従って、Notification ServiceのWindowsサービスがイベントを検知し、通知などを実行
- データ層
  - Notification Serviceのアプリケーションの構成情報を格納したデータベースインスタンス

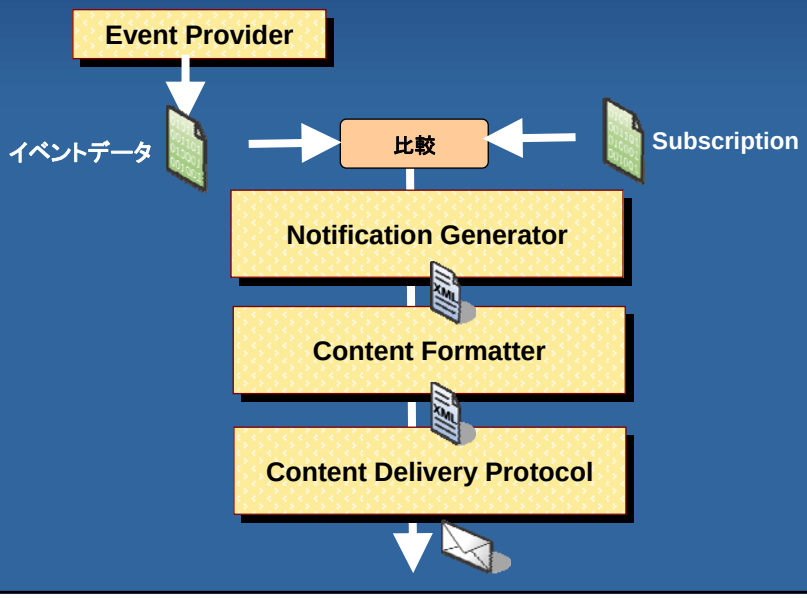
Notification Service論理構成



Notification Service論理構成

- アプリケーション構成
  - Notification Serviceアプリケーションで使用するイベントクラス、サブスクリプション、通知などのスキーマ定義
    - EventClass
    - SubscriptionClass
    - NotificationClass
- アプリケーションデータ
  - Notification Serviceアプリケーションを実行するために必要な実データを登録
    - Subscriber
    - SubscriberDevice
    - Subscription
  - ユーザアプリケーションから、Notification Serviceの.NET APIやCOMオブジェクトを利用してデータ登録

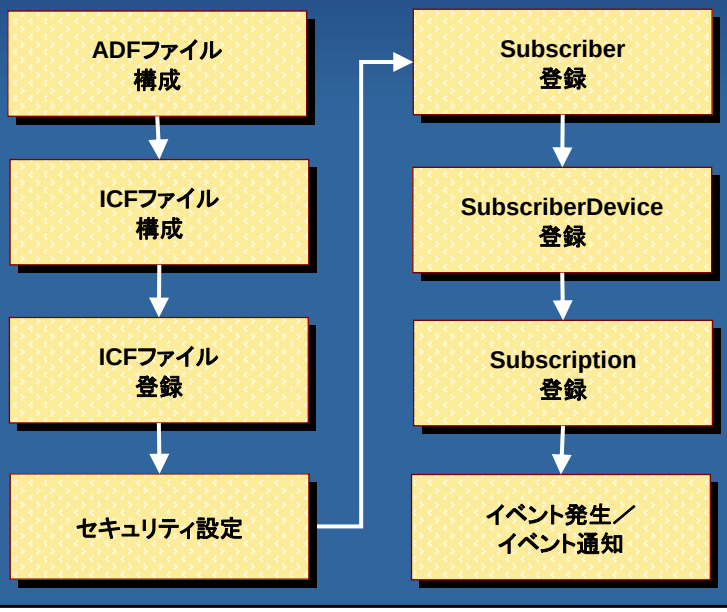
Notificationの生成



Notificationの生成

- イベントの発生をEvent Providerが検出し、イベントデータ生成
- 発生したイベントデータとSubscription(購読情報)のデータを比較し、条件が一致する購読者に対してNotificationを生成
  - NotificationはADFファイルで定義されたNotificationClassのフィールドをXML形式で保持
- Content FormatterによってNotificationが変換され、Content Delivery Protocolによって通知

構成の流れ



構成の流れ

1. Application Definition File(ADF)ファイルの作成
   - イベントをどのように検知し、通知の生成や配信をどのように行うかというNotification Servicesのアプリケーション設定
2. Instance Configuration File(ICF)ファイルの作成
   - Notification Servicesアプリケーションを実行するインスタンスの設定
3. ICFファイルの登録
   - ICFファイルをSQL Serverへ登録し、Notification Servicesのインスタンスを作成
4. セキュリティ設定
5. Subscriber登録
6. SubscriberDevice登録
7. Subscription登録
8. イベント発生／イベント通知
   - イベントを発生させ、通知を実行

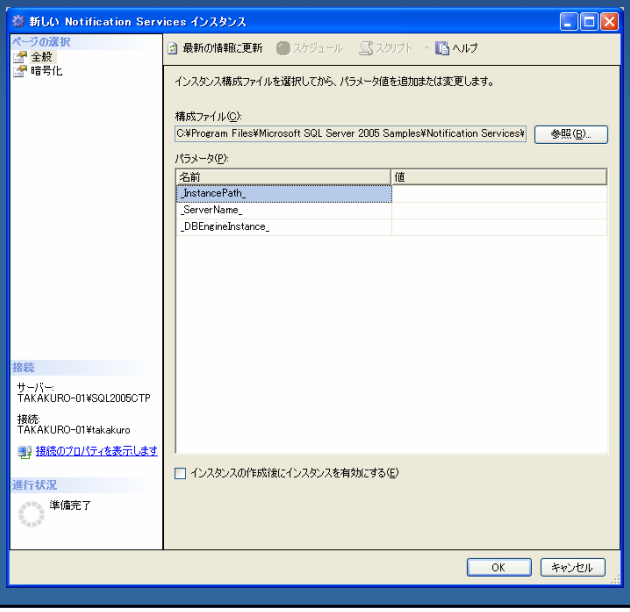
ADFファイルの構成

- アプリケーションで扱うイベントの設定
- Subscriptionの設定
- Generatorの設定
- 通知の設定
  - 利用するフォーマッタの設定
  - 利用するプロトコルの設定

ICFファイルの構成

- Notification Serviceインスタンスの設定
  - インスタンス名
- アプリケーション(ADFファイル)との関連付け
  - アプリケーション名
  - アプリケーションが利用するフォルダパス
  - ADFファイルのパス
- DeliveryChannelの登録と設定
  - 通知で利用するDelivery Channelの登録
    - SMTP
      - E-Mailでの通知
    - File
      - 通知をファイル出力
- Custom Delivery Channelを登録する場合は、Protocols要素で作成した.NETアセンブリの情報を登録
- 他にもNotification Serviceが自動で作成するデータベースインスタンスの設定などが可能

ICFファイルの登録



ICFファイルの登録

- 作成したICFファイルをSQL Server Management Studioから登録
  - ICFファイルで関連付けられたNotification Serviceアプリケーションも自動登録
- Notification Serviceで利用するデータベースインスタンスとWindowsサービスが作成される
- インスタンスの有効化／無効化や開始／停止などの管理タスクも実行

セキュリティ設定

- Windowsアカウントの作成
- Notfication Serviceが作成したデータベースインスタンスへアカウント登録
  - データベースインスタンス
    - インスタンス名＋NSMain
    - インスタンス名＋アプリケーション名
  - 追加ロール
    - NSRunService
- フォルダのアクセス権設定(Option)
  - ファイル出力などを行う場合は、出力先フォルダに対して、1.で作成したアカウントが書き込み権限を有しているか確認

Subscriberの登録

- 購読者(Subscriber)の登録
  - 名前空間
    - Microsoft.SqlServer.NotificationService
  - Subscriberクラス
    - 主要プロパティ
      - SubscriberId
        - 購読者のID
      - Enabled
        - 購読者が有効であるか

SubscriberDeviceの登録

- 購読者への通知方法(SubscriberDevice)の登録
  - 名前空間
    - Microsoft.SqlServer.NotificationService
  - SubscriberDeviceクラス
    - 主要プロパティ
      - SubscriberId
        - 購読者のID
      - DeliveryChannelName
        - 通知で利用するチャネル名
      - DeviceAddress
        - 通知先のアドレス（メールアドレスなど）
      - DeviceName
        - デバイス名

Subscriptionの登録

- 購読情報(Subscription)を登録
  - 名前空間
    - Microsoft.SqlServer.NotificationService
  - Subscriptionクラス
    - 主要プロパティ
      - SubscriberId
        - 購読者のID
      - Item
        - 購読情報として設定されたアイテム一覧
        - ADFファイルのSubscriptionClassとして定義したフィールドを利用可能

イベント発生／通知

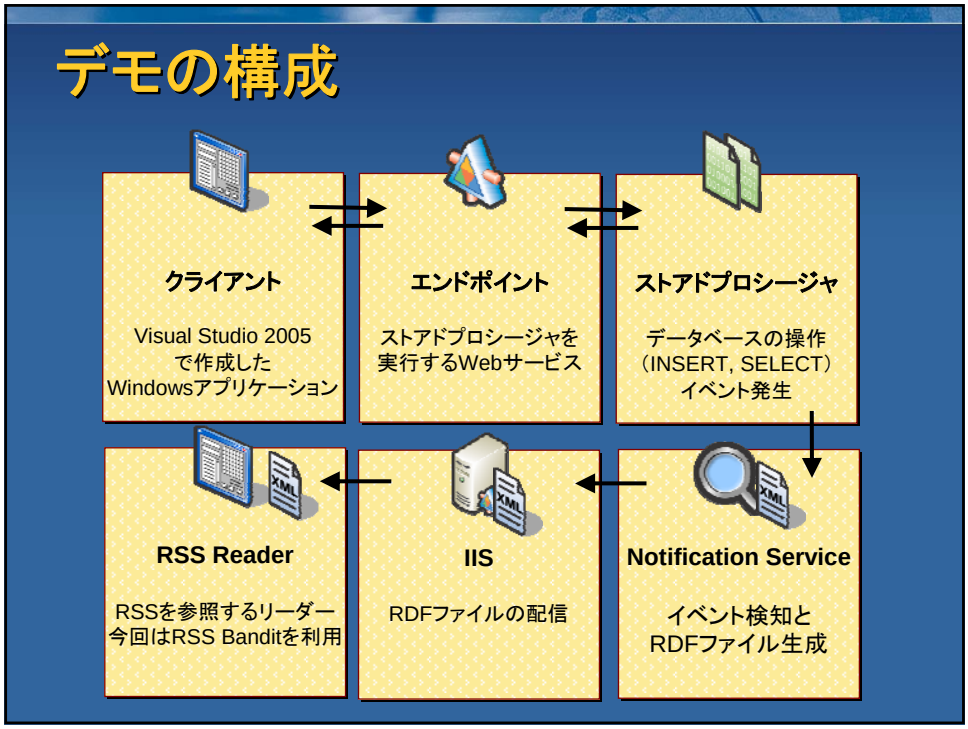
```
DECLARE @BatchID bigint;
EXEC dbo.NSEventBeginBatchWeatherEvents N'', @BatchID OUTPUT;

EXEC dbo.NSEventWriteWeatherEvents
      @EventBatchId = @BatchID,
      @City = N'Seattle',
      @Date = N'2005/02/21 14:00:00',
      @Low = 31,
      @High = 52,
      @Forecast = N'Sunny'

EXEC dbo.NSEventFlushBatchWeatherEvents @BatchID
```

イベント発生／通知

- T-SQLから直接イベントを発生
  - dbo.NSEventBeginBatch<*イベント名*>
    - イベント発生バッチ処理の開始
  - dbo.NSEventWrite<*イベント名*>
    - イベント発生
  - dbo.NSEventFlushBatch<*イベント名*>
    - イベント発生バッチ処理の終了
- データベースの変更やファイルシステムの変更イベントなどを発生
  - Custom Event Providerの作成も可能

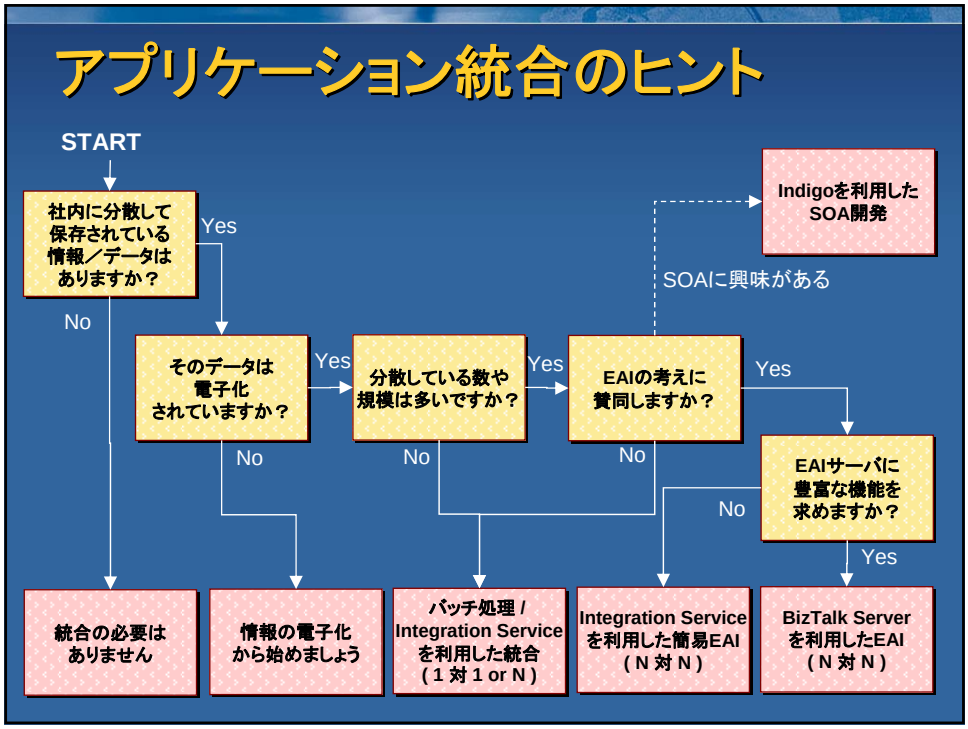
demo

RSSを利用した連携



新機能の適用箇所

- SQL Server 2005 Webサービス
  - サービス指向アーキテクチャでアプリケーションを実装する際のサービスの一つとしての利用（内部Webサービス）
  - Integration Service, BizTalk Serverなどとの連携手段
- Service Broker
  - 処理時間の長いストアドプロシージャのチューニング方法のひとつ
  - .NET APIやSQL Server 2000, MSMQとの相互運用が提供されるようになれば利用シナリオも増える
- Notification Service
  - 通知を専用で行うアプリケーションを構築する際の基盤機能（フレームワーク）として利用
  - 大規模ユーザ向き



まとめ

- 分散環境にあるデータベースを統合する技術、方法は多数ある
- 数多くあるMicrosoft技術／製品を適切な箇所、適切な用途で利用できるように、その特徴とメリット、デメリットを理解する
  - SQL Server 2005 Webサービス v.s. ASP.NET Webサービス
  - Service Broker v.s. MSMQ
  - SQL Server 2005 Integration Service v.s. BizTalk Server

Appendix

- SQL Server 2005 CTP 2005 April
  - SQL Server 2005 Books Online
  - SQL Server 2005 Samples
- A First Look at SQL Server 2005 for Developers(書籍)
  - 著者 :Bob Beauchemin, Niels Berglund, Dan Sullivan
- Archway Inc.ダウンロードページ
  - セッション資料とデモのソースコードを公開
    - http://www.archway.co.jp/download.html
- 関連セッション
  - 【ADV 1】
    - XML テクノロジー
  - 【SQL 3】
    - データ統合プラットフォーム活用シナリオ

Microsoft

Your potential. Our passion.™

Microsoft