



# これで解決、 Web サービス相互運用性 Windows と Java

**岡崎 隆之**

サン・マイクロシステムズ株式会社  
ソフトウェア・ビジネス統括本部

<http://blogs.sun.com/okazaki>



## セッション概要

Web サービスを相互運用する際の  
考慮すべき点とその解決方法  
(Project Tango) をご紹介します。

# Web サービス

を利用した相互運用を考える

システム A



システム B



システム A とシステム B を  
連携したい

システム A

システム B



TCP/IP

ネットワークケーブルを繋ぎ  
TCP/IP で通信

## システム A



## システム B



HTTP

TCP/IP

データの転送方式として  
HTTP を利用する

システム A

システム B



SOAP や REST

HTTP

TCP/IP

アプリケーション間の連携を  
Web サービスによって行う

## システム A



## システム B



### SOAP や REST

### 満たせないビジネス要件

- トランザクション
- セキュリティ
- 信頼性メッセージング
- 相互運用性

システム A

システム B



独自フレームワーク

SOAP や REST

HTTP

TCP/IP



典型的には・・・

独自フレームワークによって  
問題を解決



**コストがかかる  
相互運用性保証に時間もかかる**



# Project Tango

Web Service Interoperability Technologies (WSIT)

<https://wsit.dev.java.net>

## システム A



JAX-WS  
Project Tango(WSIT)

HTTP  
TCP/IP

## システム B



**注) JAX-WS:**

Java から Web サービスを利用するための API 仕様

## システム A

## システム B



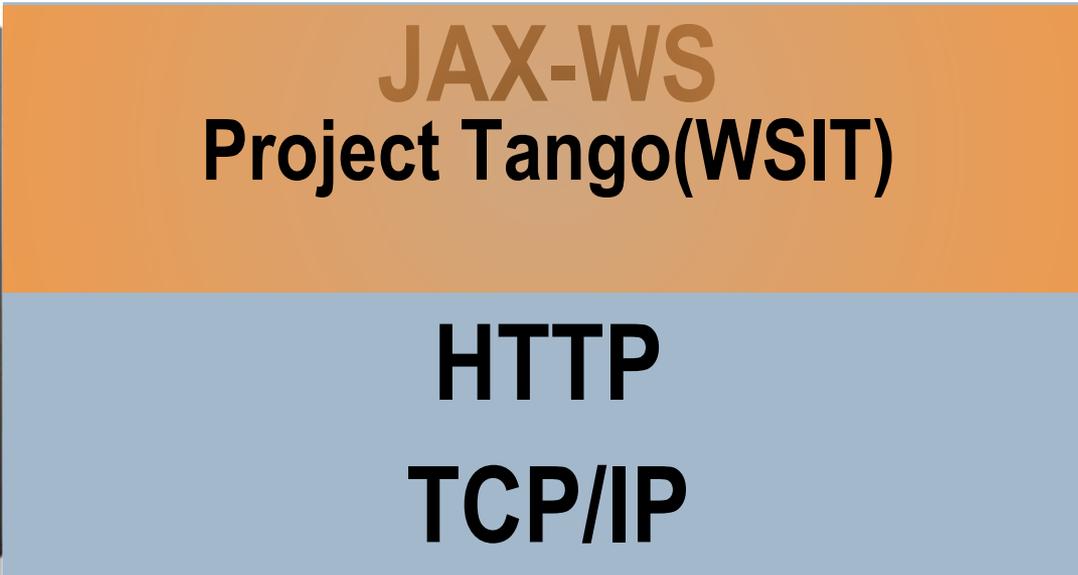
### JAX-WS Project Tango(WSIT)



**JAX-WS, SOAP, WSDL, WS-AtomicTransaction, WS-Coordination, WS-SecureConversation, WS-ReliableMessaging, WS-Trust, MTOM/XOP, WS-Security, WS-Policy, ....**

## システム A

## システム B



Project Tango は、大雑把に言うと…

**標準仕様を組み合わせた  
ランタイムライブラリです**

# システム A

# システム B



JAX-WS  
Project Tango(WSIT)

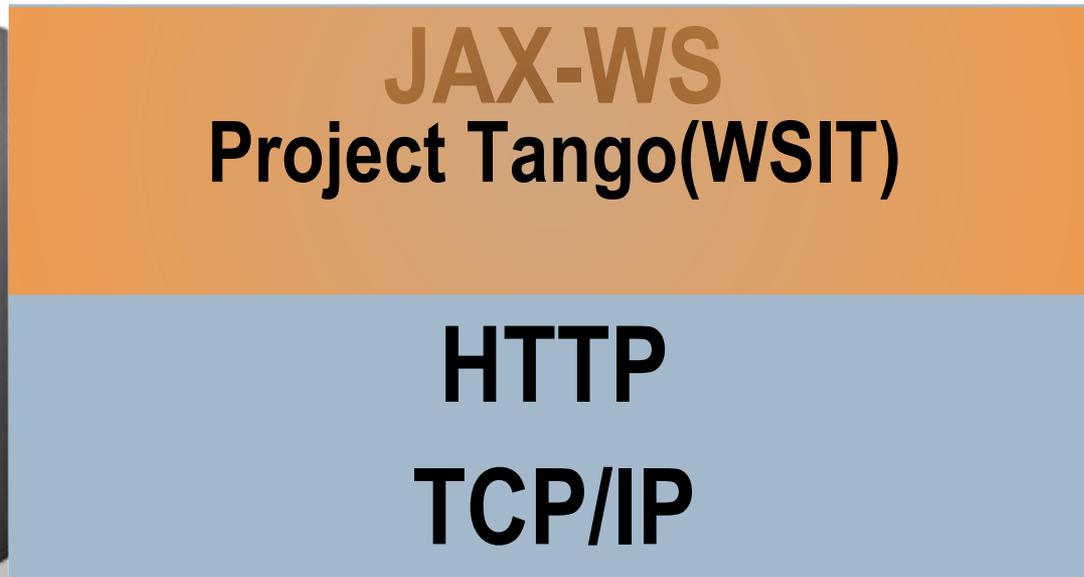
## 標準仕様を利用した実装

- SOAP
- トランザクション
- セキュリティ
- 信頼性メッセンジング
- 相互運用性 検証済み

Project Tango は、大雑  
標準仕様を  
ランタイム

## システム A

## システム B



Project Tango は、

新しいAPIではありません  
ライブラリと設定を追加するだけ

# Java アプリ



Project Tango (WSIT)

# .NET アプリ



Windows Communication Foundation (WCF)



Project Tango は、

# .NET フレームワーク (WCF) と 相互接続検証を行っています

- Project Tango (WSIT) の仕組み
  - > セキュリティ
  - > 信頼性メッセージング
  - > トランザクション
  - > ツールサポート
- Project Tango (WSIT) を使う
- Project Tango 今後の動向
- まとめ・参考資料

# Project Tango のスタック

JAX-WS ツール、NetBeans や Sun Java Studio によるサポート

セキュリティ  
(WS-SecureConv.  
WS-Trust, XWSS)

高信頼性  
(WS-Reliable  
Messaging)

トランザクション  
(WS-AtomicTransaction  
WS-Coordination)

メタデータ  
(WSDL,  
MEX  
Policy)

SOAP ベースのメッセージング  
(WSA, MTOM)

JAXB ベースの XML データバインディング (XSD, XPATH)

HTTP

TCP

SMTP

# トランスポート・プロトコル

Project Tango (WSIT) は  
HTTP や SMTP などの  
主要な普及したトランスポート  
プロトコルを利用します

HTTP

TCP

SMTP

# データ・バインディング

JAX-WS ツール

セキュリティ  
(WS-SecureConv  
WS-Trust, XWSS)

SOAP

(WSA, MTOM)

**XML/Java 間のデータ  
バインディングには  
JAXB 2.x を利用します**

**JAXB ベースの XML データバインディング (XSD, XPATH)**

HTTP

TCP

SMTP

# メッセージング

メッセージングには SOAP を  
利用します

SOAP ベースのメッセージング  
(WSA, MTOM)

JAXB ベースの XML データバインディング (XSD, XPATH)

HTTP

TCP

SMTP

# メタデータ

JAX-WS ツール、NetBeans や Sun Java Studio によるサポート

セキュリティ  
(WS-SecureConv.  
WS-Trust, XWSS)

高信頼性  
(WS-Reliable  
Messaging)

トランザクション  
(WS-AtomicTransaction  
WS-Coordination)

メタデータ  
(WSDL,  
MEX  
Policy)

WSDL や WS-Policy 等を利用して  
メタデータを交換します

SD, XPATH)

SMTP

# ビジネス要件の解決

JAX-WS ツール、NetBeans や Sun Java Studio によるサポート

**セキュリティ**  
(WS-SecureConv.  
WS-Trust, XWSS)

**高信頼性**  
(WS-Reliable  
Messaging)

**トランザクション**  
(WS-AtomicTransaction  
WS-Coordination)

メタデータ  
(WSDL,  
MEX  
Policy)

SOAP ベースのメッセージング

**各種ビジネス要件を満たすため  
様々な Web サービス仕様を利用します**

# ツールによるサポート

JAX-WS ツール、NetBeans や Sun Java Studio によるサポート

セキュリティ

高信頼性

トランザクション

生産性を向上させるため  
様々なツールによるサポートを提供します

JAXB ベースの XML データバインディング (XSD, XPATH)

HTTP

TCP

SMTP

- Project Tango (WSIT) の仕組み
  - > セキュリティ
  - > 信頼性メッセンジング
  - > トランザクション
  - > ツールサポート
- Project Tango (WSIT) を使う
- Project Tango 今後の動向
- まとめ・参考資料

# セキュリティ

- WS-Security
  - > メッセージの完全性、機密性、単一メッセージの認証による保護品質の提供
- WS-SecureConversation
  - > 複数のメッセージをやり取りする際の安全なセキュリティ・コンテキスト・トークン
- WS-SecurityPolicy
  - > Web サービスのセキュリティ・ポリシーを定義
- WS-Trust
  - > クライアント側とサーバ側のセキュリティトークンの要求、発行および信頼関係の管理

# WS-Security: 完全性や機密性

- 完全性
  - > メッセージが転送中に改ざんされていないことを保証
- 機密性
  - > 目的の受信先のみメッセージが転送されることを保証
- 送信元の認証
  - > メッセージの送信元を識別する

# WS-SecureConversation

- セキュリティ・コンテキスト・トークンを利用して、SecureConversation のセッションを作成・管理し、セッション内におけるメッセージ交換の保護

クライアント

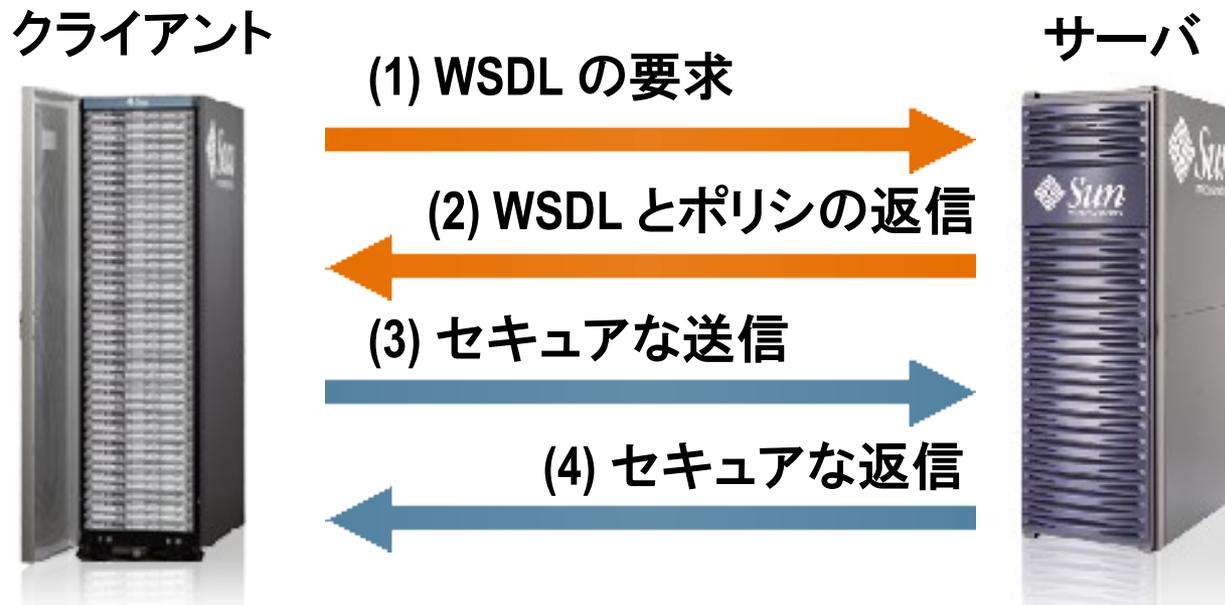


サーバ



# WS-SecurityPolicy: ポリシー

- Web サービスごとのセキュリティ・ポリシーの定義と、ポリシー情報の交換
  - > 例: 個人情報を取り扱う Web サービスはセキュリティ強度の高い認証・通信方式を利用する

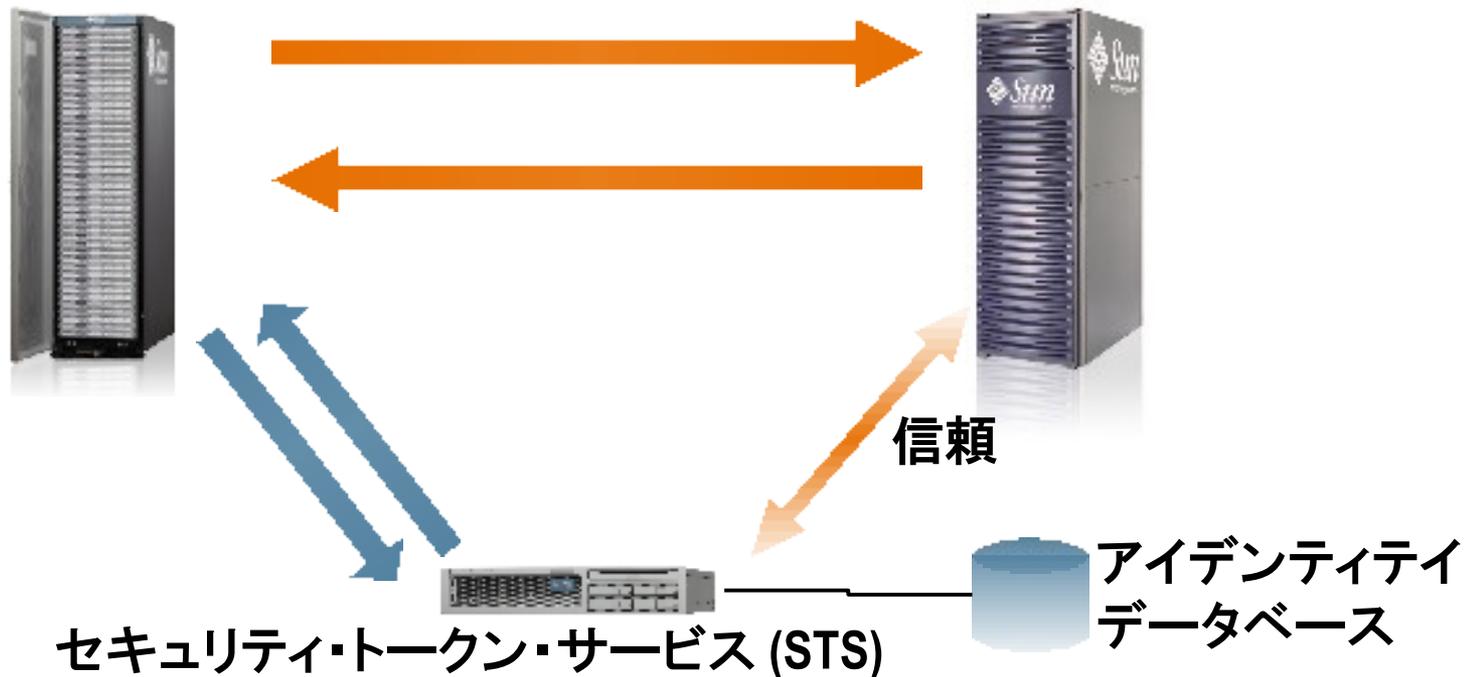


# WS-Trust : 信頼関係の管理

- クライアント・サーバ間の信頼関係の管理
- セキュリティ・トークンの要求、発行

クライアント

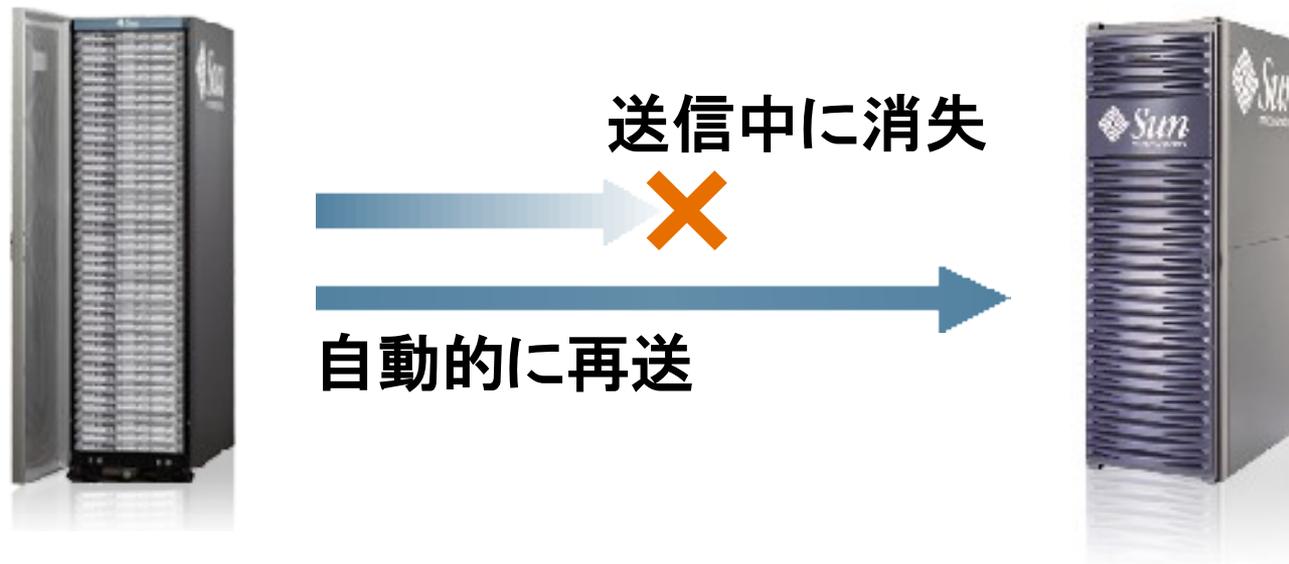
サーバ



- Project Tango (WSIT) の仕組み
  - > セキュリティ
  - > 信頼性メッセンジング
  - > トランザクション
  - > ツールサポート
- Project Tango (WSIT) を使う
- Project Tango 今後の動向
- まとめ・参考資料

# 信頼性メッセージング

- WS-ReliableMessaging
  - > 送信中に失われたメッセージの再送処理



- > 順番どおりにメッセージが送信されることを保証
- > 少なくとも1回送信、多くとも1回送信などの保証

- Project Tango (WSIT) の仕組み
  - > セキュリティ
  - > 信頼性メッセンジング
  - > トランザクション
  - > ツールサポート
- Project Tango (WSIT) を使う
- Project Tango 今後の動向
- まとめ・参考資料

# トランザクション

- トランザクション化された Web サービスのサポート
  - アトミック性 (Atomicity)、一貫性 (Consistency)、独立性 (Isolation)、永続性 (Durability) の保証
- WS-Coordination
  - 分散アプリケーションの動作を調整し、分散トランザクションの結果が矛盾なく一致するようにする
- WS-AtomicTransaction
  - ACID トランザクションを Web サービスに適用する
- アプリケーションサーバのトランザクション・マネージャを利用
- トランザクション利用のために新しい API を覚える必要はありません

# アノテーションによる指定

## トランザクションをサポートする Web サービスの実装

```
@javax.jws.WebService
@javax.ejb.Stateless
@javax.ejb.TransactionManagement (CONTAINER) [1]

public class Bank {

    @javax.jws.WebMethod
    @javax.ejb.TransactionAttribute (REQUIRED) [1] [2]
    void transferFunds (...) throws ... ;
}
```

[1] ステートレス EJB のデフォルト、ここでは明示的に宣言しています

[2] 実装上の制限により One Way メッセージではトランザクションは  
利用できません

# Transacted Web サービスの定義

```
<wsdl:definitions>
  <!-- WS-AT ポリシ・アサーションの定義 -->
  <wsp:Policy wsu:Id="TransactedPolicy1" >
    <wsat:ATAssertion wsp:Optional="true"/>
    <wsat:ATAlwaysCapability/>
  </wsp:Policy>
  <wsdl:binding name="Bank" type="tns:BankPortType" >
    <!-- ポリシーによってトランザクションサポートに -->
    <wsdl:operation name="transferFunds" >
      <wsp:PolicyReference URI="#TransactedPolicy1"
        ...
      </wsdl:operation>
    </wsdl:binding>
  </wsdl:definitions>
```

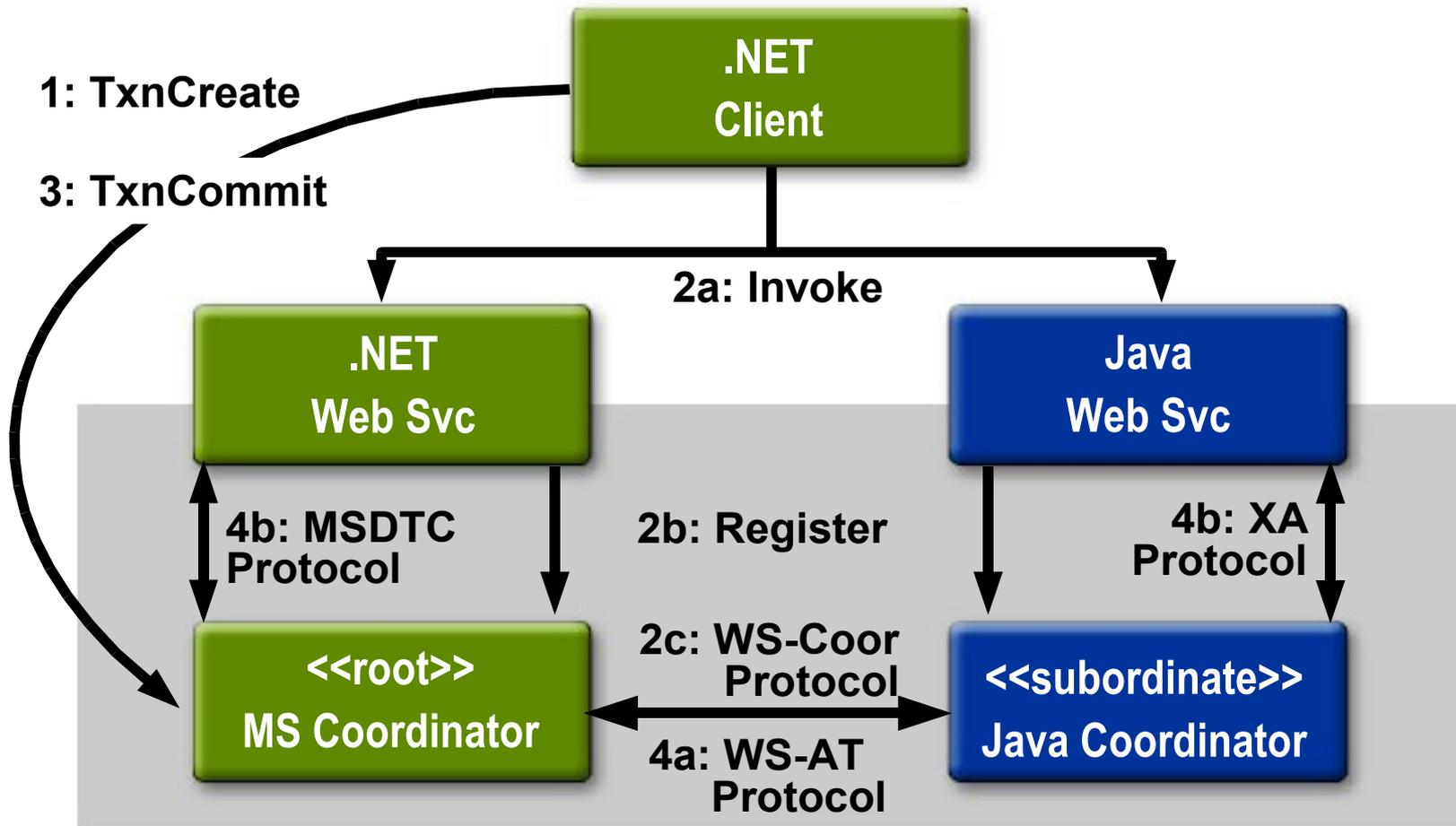
# クライアント側のトランザクション

```
import javax.annotation.Resource;  
  
public class ATMClient {  
    @javax.jws.WebServiceRef  
    static BankService service;  
  
    public void selectedTransferFunds() {  
        @Resource javax.transaction.UserTransaction ut;  
  
        Bank bank = service.getBank();  
        ut.begin();  
        bank.transferFunds( ... );  
        ut.end();  
    }  
}
```

**JTA(Java Transaction API) の UserTransaction を使って  
トランザクション境界をコントロール**

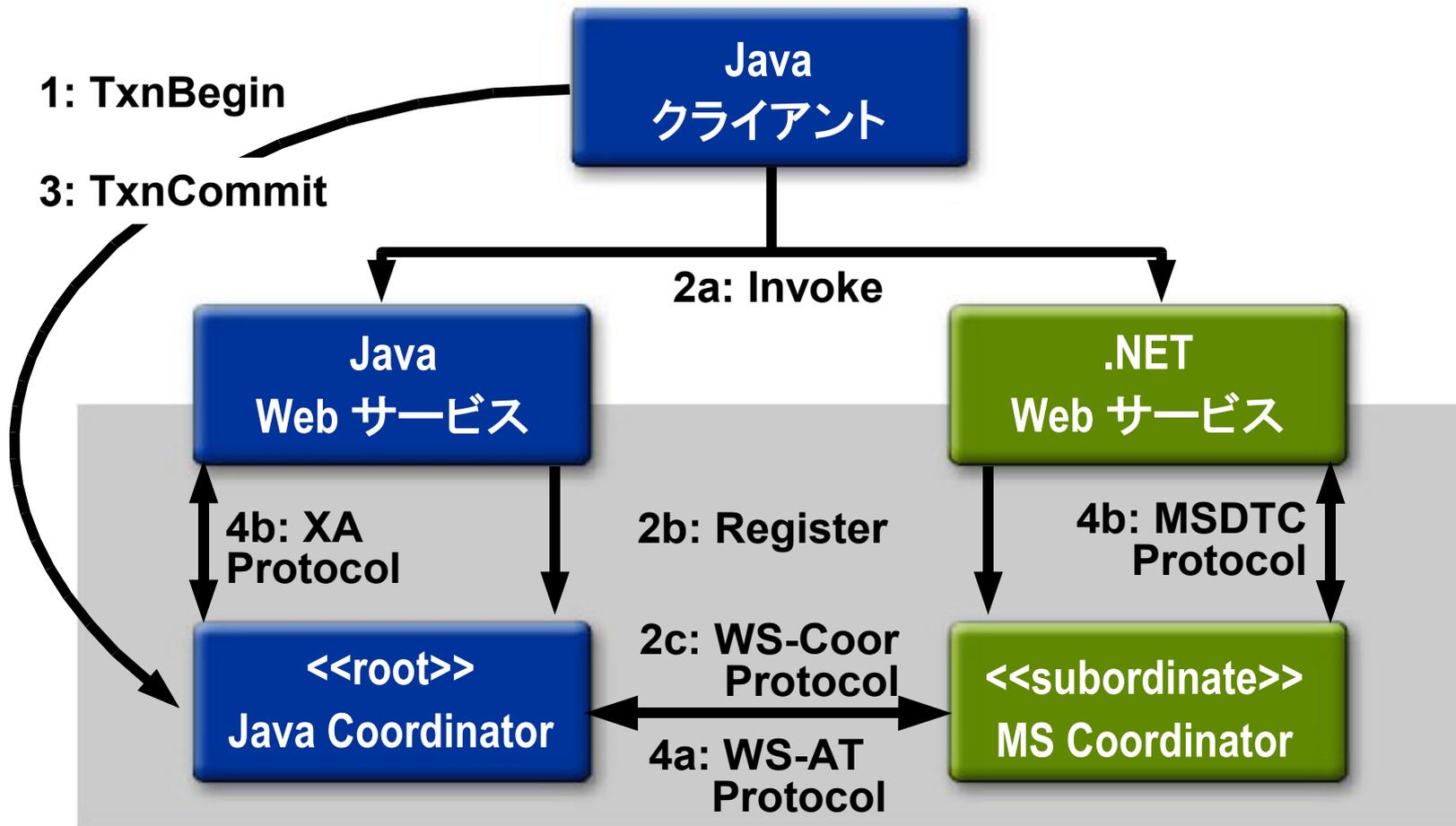
# 分散トランザクション (1)

.NET クライアント主導の分散トランザクションの解決



# 分散トランザクション (2)

Java EE/WSIT クライアント主導の分散トランザクションの解決



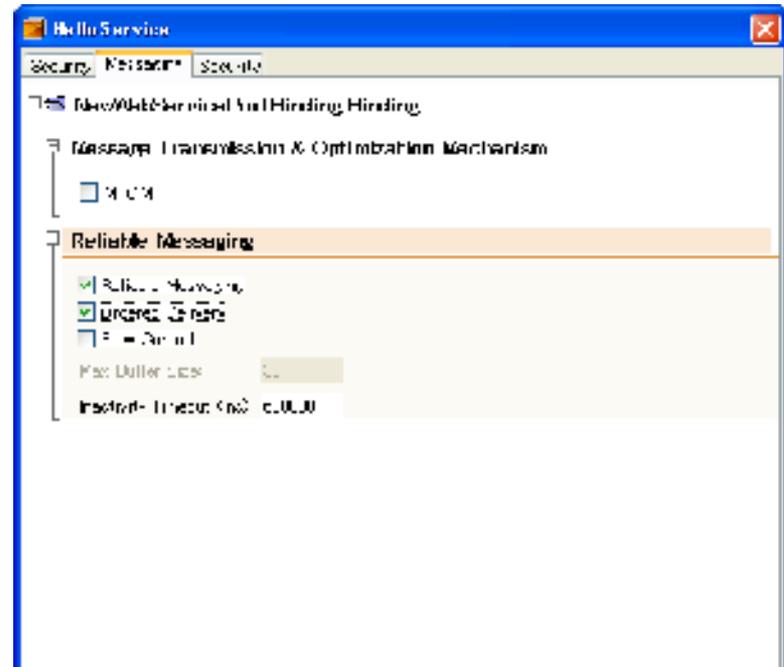
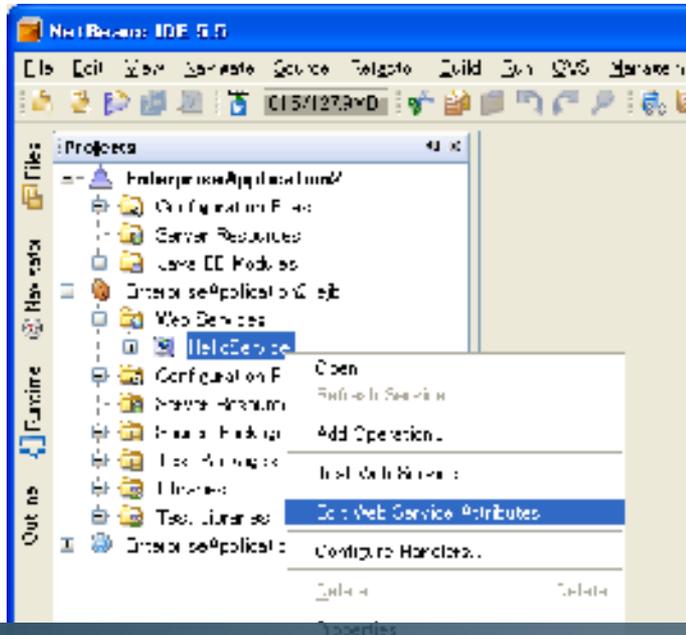
- Project Tango (WSIT) の仕組み
  - > セキュリティ
  - > 信頼性メッセンジング
  - > トランザクション
  - > ツールサポート
- Project Tango (WSIT) を使う
- Project Tango 今後の動向
- まとめ・参考資料

# WSIT を利用する開発形態



WSIT の設定ファイルを作成・パッケージ化し  
WSIT 対応のアプリケーションサーバに  
アプリケーションを配置する

# NetBeans WSIT プラグイン

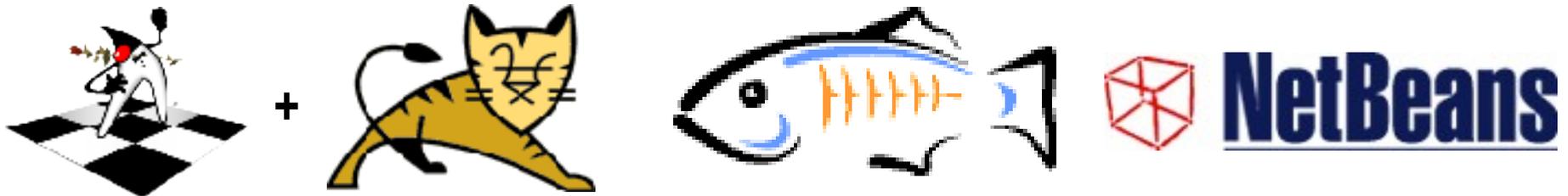


高信頼性メッセージングなどに関する設定を作成した Web サービスに対してダイアログから設定

- Project Tango (WSIT) の仕組み
  - > セキュリティ
  - > 信頼性メッセンジング
  - > トランザクション
  - > ツールサポート
- Project Tango (WSIT) を使う
- Project Tango 今後の動向
- まとめ・参考資料

# Project Tango を使うために必要な物

- Project Tango(WSIT) の利用のために必要なもの
  - > GlassFish v2 または Tomcat 5.5
  - > Apache Ant 1.6.5 以上
  - > Project Tango(WSIT) の配布ファイル<sup>[1]</sup>
  - > NetBeans 5.5 と WSIT プラグイン<sup>[2]</sup>



[1] WSIT は GlassFish v2 リリース時に GlassFish に含まれた形で提供されるようになりますが、現在は別々に配布されています

[2] WSIT プラグインはアップデートセンタ経由で配布されています

# Project Tango を使うための準備 (1)

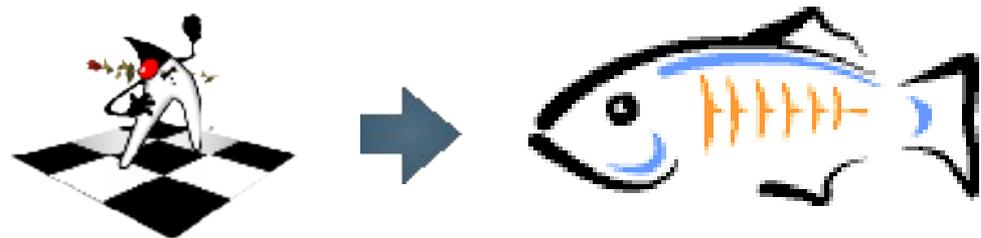
- GlassFish の入手
  - > GlassFish v2 Main branch b15<sup>[1]</sup> の入手
- GlassFish のインストール
  - > `java -Xmx256m -jar glassfish-installer-v2-b15-platform.jar`
- GlassFish のセットアップ
  - > 展開された配布ファイル内の設定ファイルを使ってセットアップ
  - > `ant -f setup.xml`



[1] [https://glassfish.dev.java.net/public/downloadsindex.html#Promoted\\_binary\\_builds](https://glassfish.dev.java.net/public/downloadsindex.html#Promoted_binary_builds)

# Project Tango を使うための準備 (2)<sup>[1]</sup>

- WSIT ファイルの入手
  - > <https://wsit.dev.java.net> よりバイナリを入手 <sup>[2]</sup>
  - > 入手した jar ファイルを実行しファイルを展開
    - > `java -jar wsit-1_0_M20060827-fcs-bin-b03c-14_sep_2006.jar`
- GlassFish にインストール
  - > 展開した WSIT 配布ファイルに含まれる ant スクリプトを利用
  - > `ant -Das.home=[AS のインストール場所] -f wsit-on-glassfish.xml install`



[1] このページの手順は GlassFish v2 リリース後には不要となる手順です

[2] <https://jax-ws.dev.java.net/servlets/ProjectDocumentList?folderID=5648&expandFolder=5648&folderID=5648>

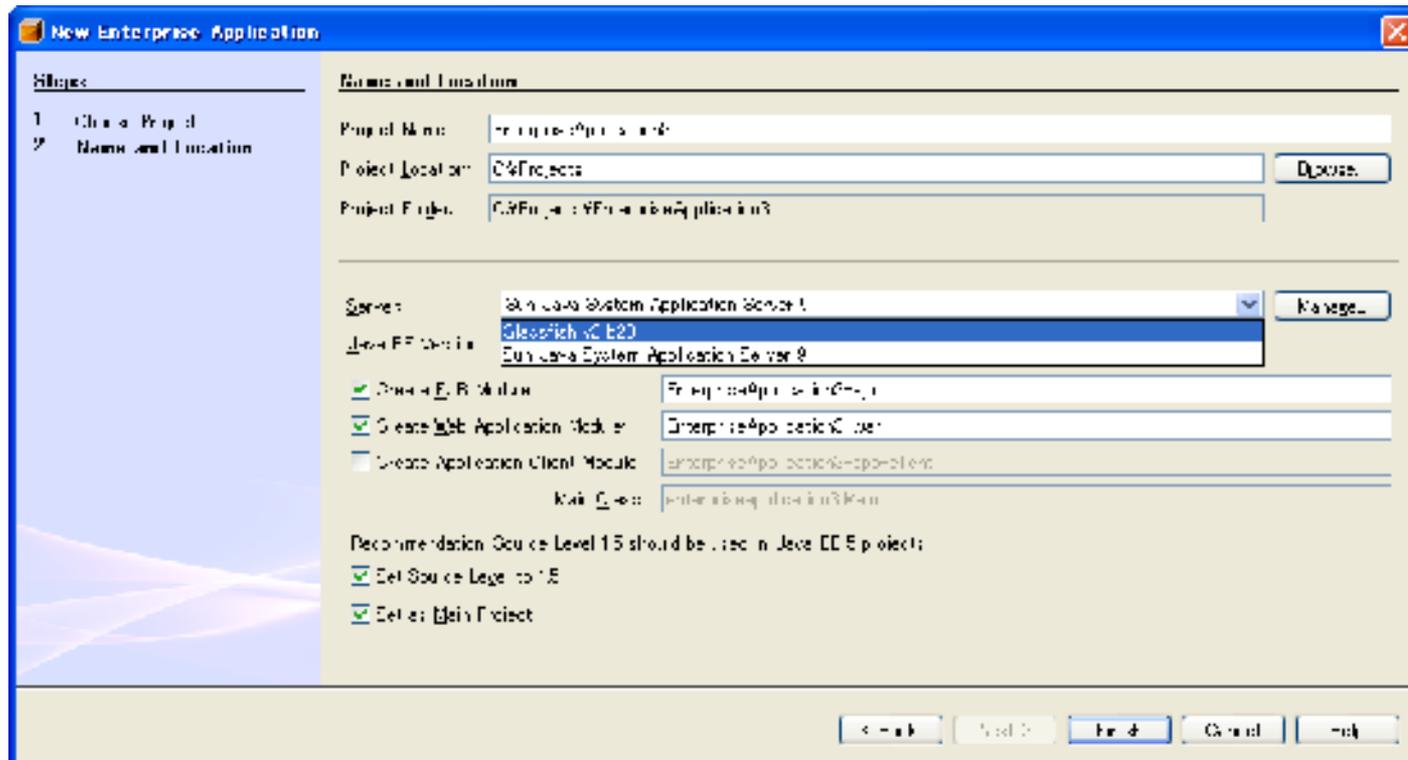
# Project Tango を使うための準備 (3)

- NetBeans の入手とインストール
  - > <http://jp.netbeans.org> より入手しインストール
- WSIT プラグインのインストール
  - > メニューの「ツール」→「アップデートセンター」より WSIT プラグインを入手、インストール
- GlassFish を開発環境用サーバとして登録
  - > 「実行時」ウィンドウの「サーバ」よりサーバ登録



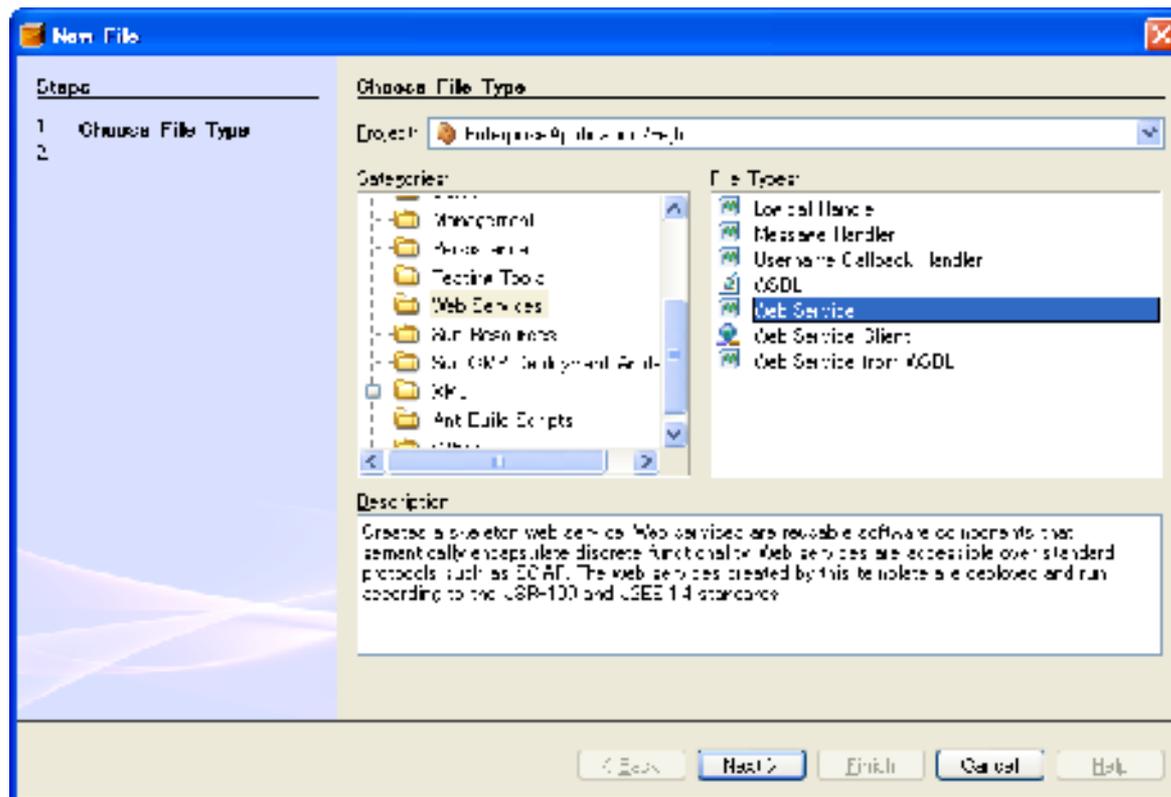
# Web サービス作成と WSIT の設定 (1)

- NetBeans 上にプロジェクトを作成
  - > プロジェクトの実行環境として Glassfish v2 を選択



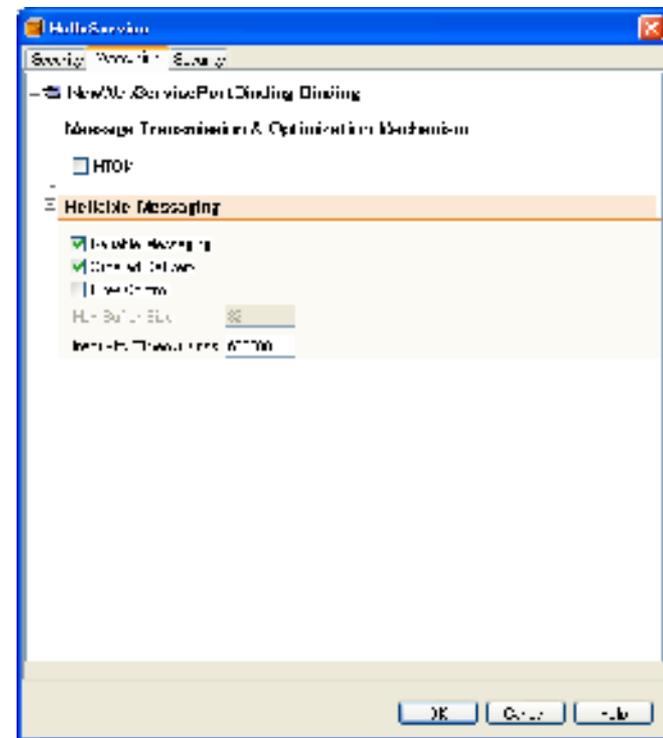
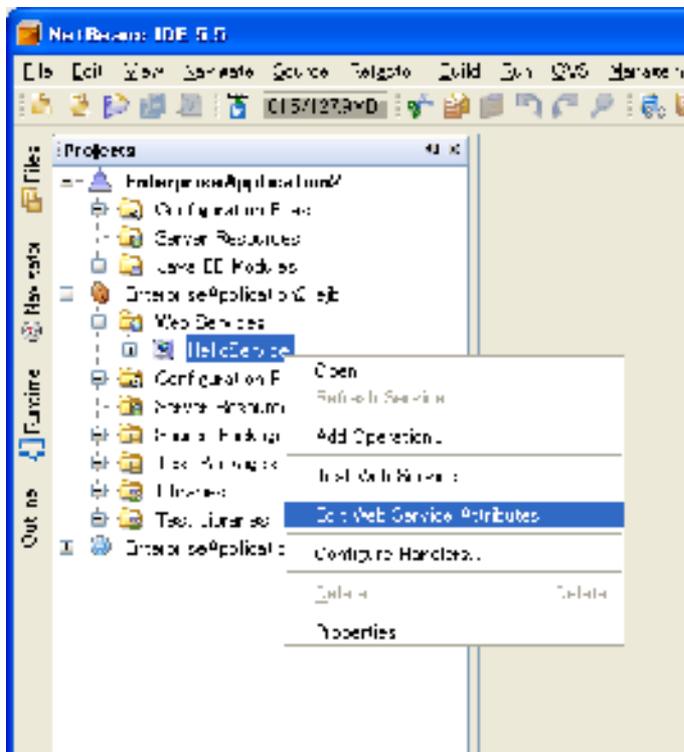
# Web サービス作成と WSIT の設定 (2)

- Web サービスの作成
  - > メニュー「新規作成」から Web サービスを選択



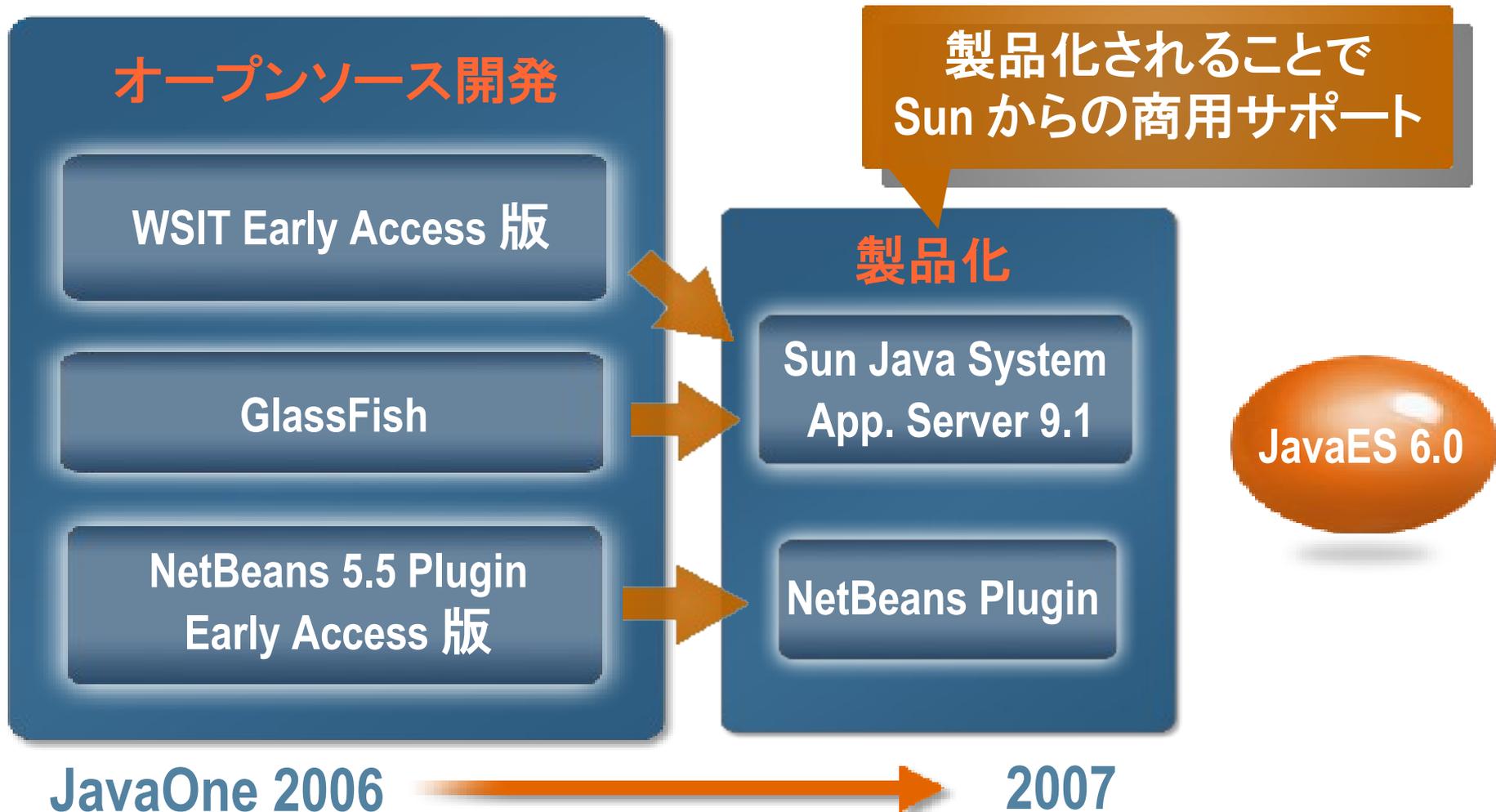
# Web サービス作成と WSIT の設定 (3)

- WSIT の設定
  - > 作成した Web サービスを選択、右クリックし「Edit Web Service Attributes」を選ぶ



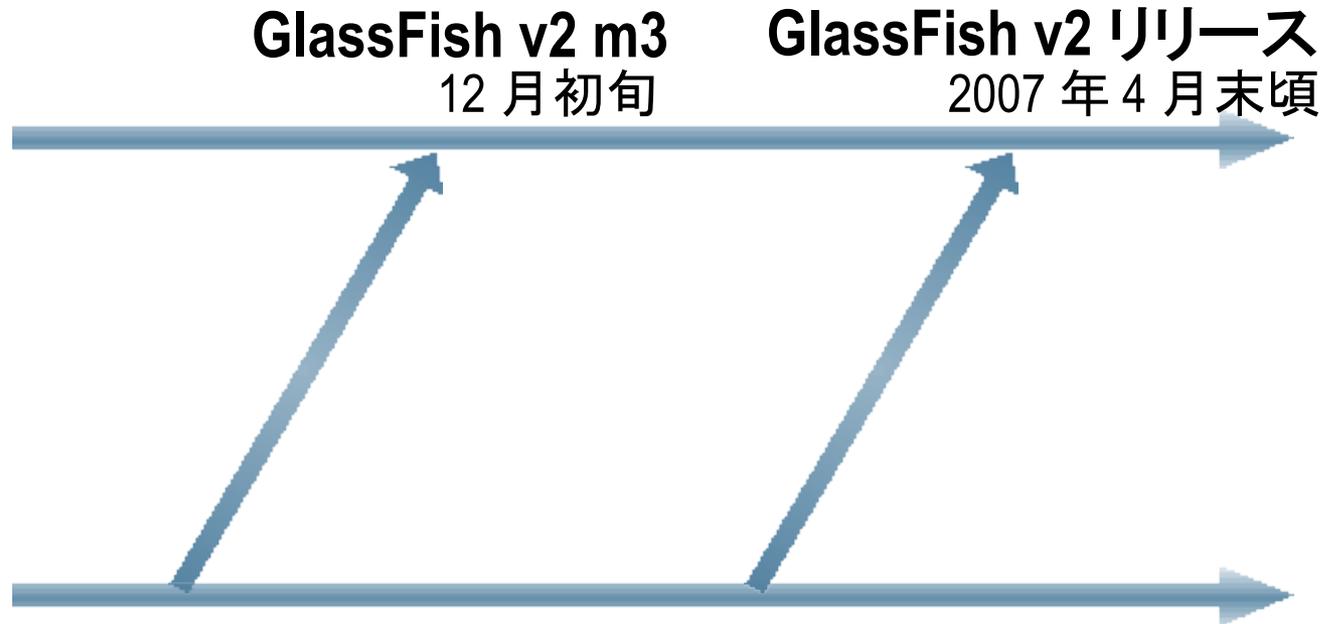
- Project Tango (WSIT) の仕組み
  - > セキュリティ
  - > 信頼性メッセンジング
  - > トランザクション
  - > ツールサポート
- Project Tango (WSIT) を使う
- Project Tango 今後の動向
- まとめ・参考資料

# Project Tango のロードマップ



注：ロードマップは予告なく変更される場合があります

# GlassFish と Project Tango



JavaOne 2006



2007

注：ロードマップは予告なく変更される場合があります

GlassFish の最新ロードマップ : <http://www.glassfishwiki.org/gfwiki/Wiki.jsp?page=Milestones>

- Project Tango (WSIT) の仕組み
  - > セキュリティ
  - > 信頼性メッセンAGING
  - > トランザクション
  - > ツールサポート
- Project Tango (WSIT) を使う
- Project Tango 今後の動向
- まとめ・参考資料

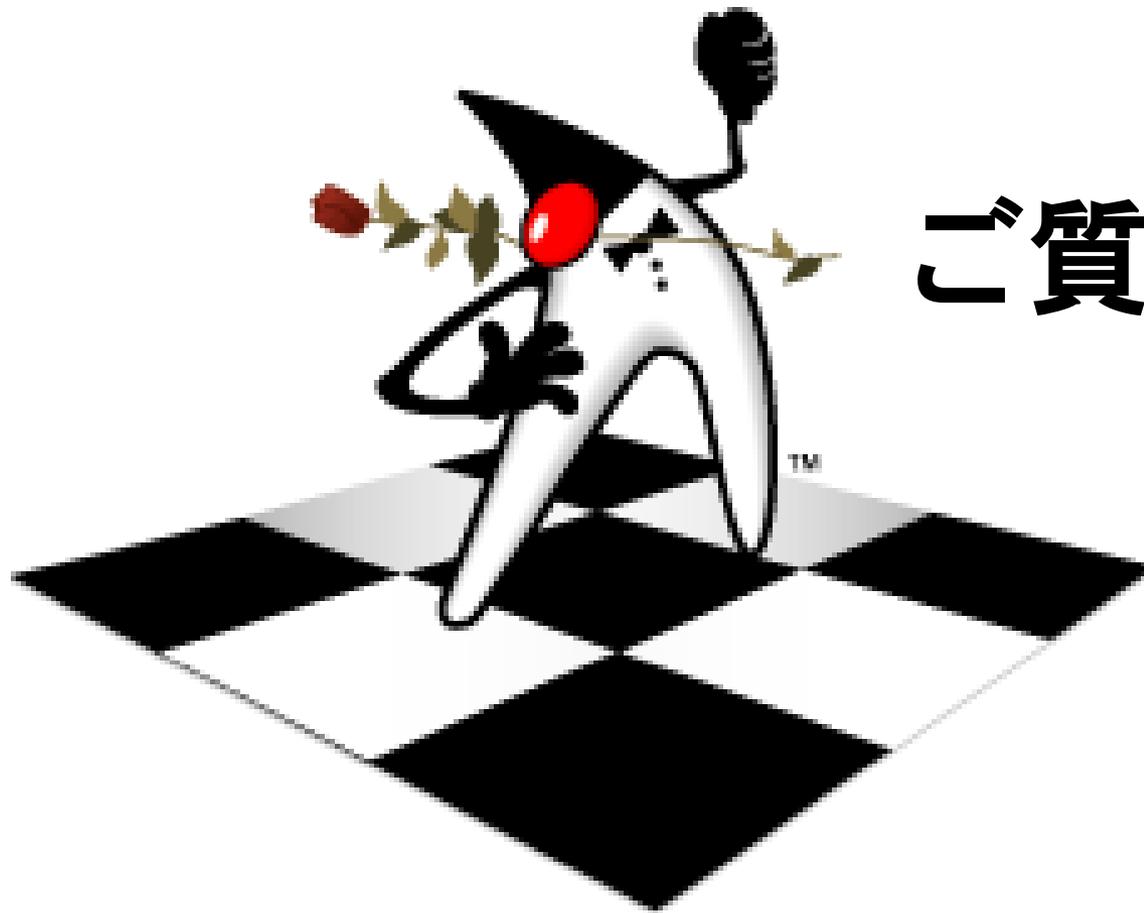
## まとめ

- Project Tango(WSIT) は Web サービス相互運用性の問題を解決するオープンソース・プロジェクトです
- Project Tango(WSIT) は標準ベースの技術です
- Project Tango(WSIT) を利用する際にプログラムの変更は必要ありません

# 参考資料

- 関連サイト
  - > <https://wsit.dev.java.net> - Project Tango Web サイト
  - > <https://glassfish.dev.java.net> - Project GlassFish Web サイト
  - > <http://msdn.microsoft.com/webservices/indigo/> - Project Indigo(WCF)
- チュートリアル
  - > <http://java.sun.com/webservices/interop/reference/tutorial/doc/index.html>
- Web サービス相互互換性ポータル
  - > <http://java.sun.com/webservices/interop/>
  - > <http://msdn.microsoft.com/webservices/building/interop>
  - > <http://mssoapinterop.org/ws/>

ご静聴ありがとうございました



ご質問をどうぞ



# これで解決、 Web サービス相互運用性 Windows と Java

**Takayuki Okazaki**

[takayuki.okazaki@sun.com](mailto:takayuki.okazaki@sun.com)

<http://blogs.sun.com/okazaki>

