



～ 第8回 XMLコンソーシアムDay ～ sPlatプロジェクト活動報告

ポリシーの伝達について

2006年12月12日
アドソル日進株式会社
荒本道隆



sPlatの検討状況



- 現在、マルチホップのWebサービス呼び出しにおける、特に中継者で発生する課題を検討中
 - WS-Securityの最大のメリットは、End-to-Endのセキュリティ
 - WS-Security関連のミドルウェアは、署名・暗号化したり、署名検証・復号する機能は提供してくれるけど、中継する機能がない
- サイト間のポリシーの伝達方法については、まだまだ未検討

ポリシーの種類

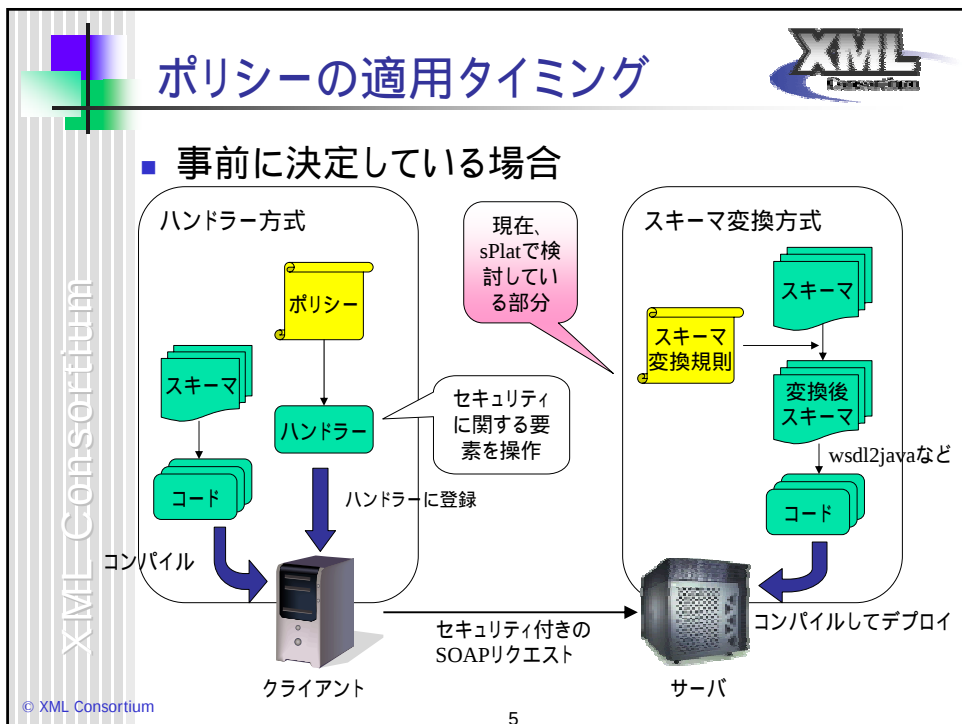


- WS-Policy
 - WS-SecurityPolicy (セキュリティ関連)
 - WS-RM Policy (高信頼メッセージング関連)
 - WS-BA Policy (WS-BusinessActivity)
 - WS-AtomicTransaction
 - WS-MTOMPolicy (メッセージ送信の最適化関連)
- WS-MetadataExchange (メタデータ取得関連)

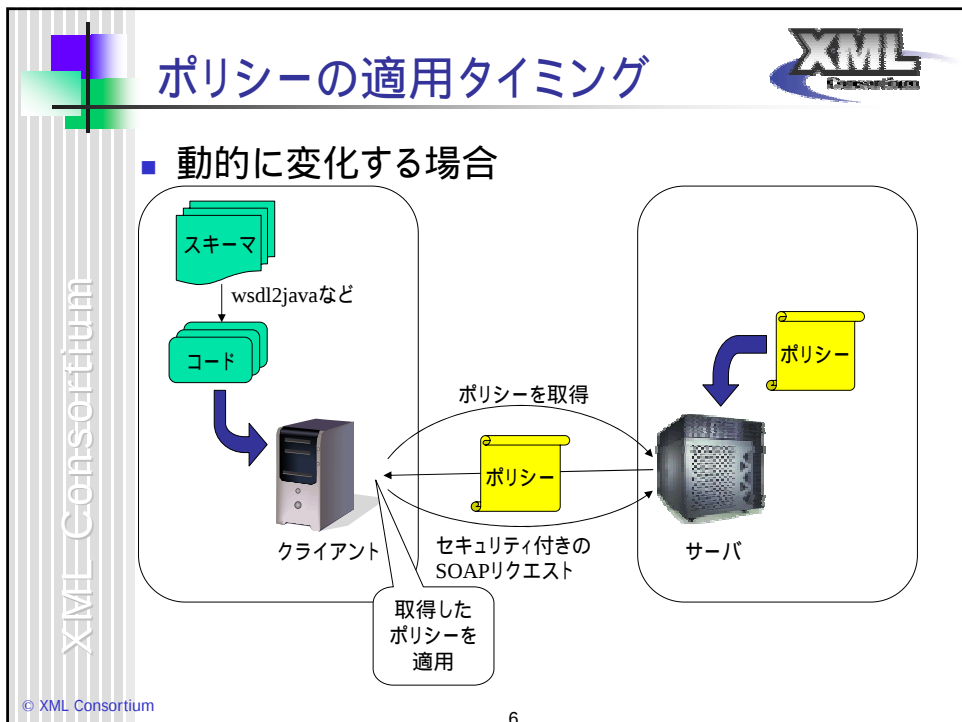
ポリシーの分類



- 事前に決定
 - 事前に何らかの方法でポリシーが伝達できていればいい
 - 変更時は、アプリの再起動などを伴う
 - 事前コンパイルを伴う方式(スキーマ変更方式など)が使える
 - ポリシーの異なる相手が複数ある場合、それぞれに対応が必要
- 動的に変化
 - 動的なポリシーの伝達方法が必要
 - ポリシーの異なる相手が複数あっても、呼び出す毎にポリシーを取得する事で対応できる



5



6



WS-Policyの実装例の紹介



XML Consortium

- WSIT(Web Services Interoperability Technologies)
 - WS技術の相互接続性を実現するための活動
 - MicrosoftのWCF(Windows Communication Foundation)との相互運用の確立

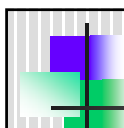


XML Consortium

- <http://java.sun.com/webservices/interop/>
- 開発者向けサイト <https://wsit.dev.java.net/>
 - テストケースのダウンロード
 - 開発者サイトの Source から /wsit/wsiti/test/e2e/testcases/wssc/

© XML Consortium

7

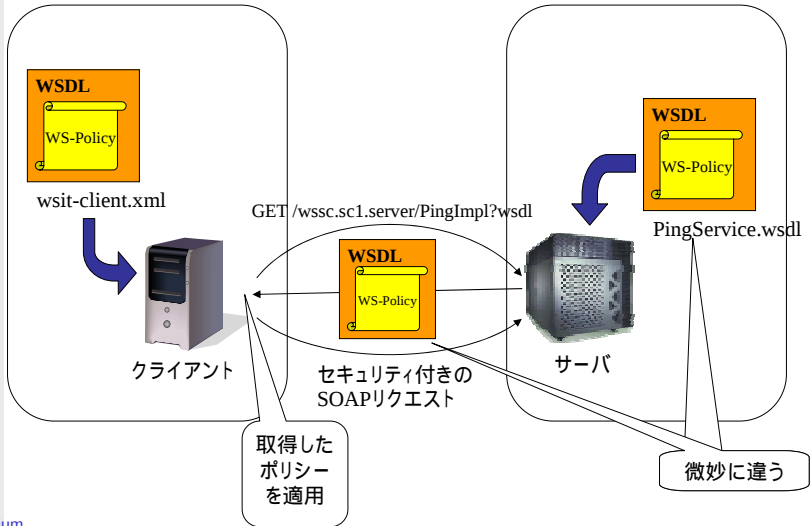


WSITのポリシーの交換方法



XML Consortium

- テストケース wssc/sc1 の場合



© XML Consortium

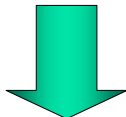
8



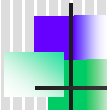
動的にポリシーが変化する場合



- Point-to-Pointのポリシーの伝達
 - WSITのような方式で対応可能
- End-to-Endのポリシーの伝達
 - 直接ポリシーを取得するのか？
 - 中継者が仲介するのか？
 - 中継者でどうやってマージするのか？



今後の検討課題



付録



WSITのテストケースwssc/sc1の動作手順 (2006/11/25版)



XML Consortium

- WSITのテストケースwssc/sc1の動作手順
 - 11/25の段階では、うまくテストケースが動かなかった
 - 以下は、なんとかテストケースを動かしてみる方法で、**正式な手順ではありません**
 - Windows上でTOMCATを使った場合の手順
- それぞれのパスを環境変数にセット(D:\WORKの下に環境構築した場合)
 - SET WSIT_HOME=D:\WORK
 - SET JAVA_HOME=C:\Java\jdk1.5.0_09
 - SET ANT_HOME=C:\Apache-ant-1.6.5
 - SET PATH=%JAVA_HOME%\bin;%ANT_HOME%\bin;%PATH%
 - SET CATALINA_HOME=%WSIT_HOME%\apache-tomcat-5.5.20
 - SET CATALINA_OPTS=-DWSIT_HOME=%WSIT_HOME%
- CVSからwsit/wsit/をダウンロード
 - https://wsit.dev.java.net/source/browse/*checkout*/wsit/wsit/docs/howto/WSIT_Download_Build_Install.html
- 使用するキーストアをダウンロード
 - <http://java.sun.com/webservices/interop/reference/tutorial/doc/Security3.html#wp125758>
 - %WSIT_HOME%\xws-security\etc\ の下に4つのキーストアを置く
- WSITをビルド
 - cd %WSIT_HOME%\wsit\wsit
 - ant clean main
- TOMCATにWSITをインストール
 - cd %WSIT_HOME%\wsit\wsit\dist\image\jax-ws-latest-wsit
 - ant -f wsit-on-tomcat.xml install
- TOMCATにデプロイ用のロールとユーザーを追加
 - notepad.exe %CATALINA_HOME%\conf\tomcat-users.xml

```
<role rolename="manager" />
<user username="admin" password="admin" roles="manager" />
```

詳細は、WSITのWeb
サイトや付属のドキュメントを参照

© XML Consortium

11



WSITのテストケースwssc/sc1の動作手順 (2006/11/25版)



XML Consortium

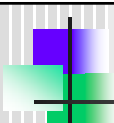
- TOMCATを起動
 - cd %CATALINA_HOME%\bin
 - startup.bat
- tcpmonを起動(通信内容を確認するため)
 - <http://www.apache.org/dist/ws/commons/tcpmon/1.0/>
 - 解凍して tcpmon-1.0-bin\build\tcpmon.bat を起動
 - 8888ポートで受けて、localhost:8080へ
- テストケースの設定ファイルを修正(alias名はkeytoolで確認)
 - cd %WSIT_HOME%\wsit\wsit\test\testcases\wssc\sc1
 - notepad.exe resources\wsit-client.xml

```
<sc:KeyStore wssp:visibility="private" location="$WSIT_HOME/xws-security/etc/client-keystore.jks" type="JKS" alias="xws-security-client" storepass="changeit"></sc:KeyStore>
<sc:TrustStore wssp:visibility="private" location="$WSIT_HOME/xws-security/etc/client-truststore.jks" type="JKS" storepass="changeit" peeralias="alias" sstalias="certificate-authority"></sc:TrustStore>
```

- テストケースwssc/sc1をTOMCATを使って実行
 - cd %WSIT_HOME%\wsit\wsit\test\test\testcases\wssc\sc1
 - java -jar %WSIT_HOME%\wsit\wsit\test\test\lib\harness.jar -tomcat-remote admin:admin@localhost:8888 testcases\wssc\sc1
 - ここではWebサービス呼び出し時にセキュリティに関する物が何も付かないので「ERROR: No security header found in the message」が発生する
 - サーバとクライアントのコードが自動生成されればOK

© XML Consortium

12



WSITのテストケースwssc/sc1の動作手順 (2006/11/25版)



XML Consortium

- TOMCAT に wssc/sc1 をデブロイ (テストケース実行後にアンデブロイされるため)
 - cd %WSIT_HOME%\wsit\test\testcases\wssc\sc1\work
 - copy services\server\wssc.sc1.server.war %CATALINA_HOME%\webapps
- クライアントコードを記述する
 - cd %WSIT_HOME%\wsit\test\testcases\wssc\sc1\work
 - notepad.exe client-source\wssc\sc1\client\TestRun.java

```
package wssc.sc1.client;
import javax.xml.ws.Holder;

public class TestRun {
    public static void main(String[] args) {
        try {
            PingService port = new PingService().getPingPort();
            port.ping(new Holder("1"), new Holder("sun"), new Holder("Passed!"));
        } catch (Exception ex) {
            System.err.println (ex);
        }
    }
}
```

- クライアントをコンパイルして実行
 - SET CLASSPATH=.:%WSIT_HOME%\wsit\dist\image\jax-ws-latest-wsit\lib\webservices-rt.jar;%WSIT_HOME%\wsit\dist\image\jax-ws-latest-wsit\lib\webservices-tools.jar
 - cd %WSIT_HOME%\wsit\test\testcases\wssc\sc1\work
 - javac -d client-classes client-source\wssc\sc1\client*.java
 - cd client-classes
 - copy ..\..\resources\wsit-client.xml .
 - Java -DWSIT_HOME=%WSIT_HOME% wssc.sc1.client.TestRun

© XML Consortium
13



XML Consortium

ご清聴ありがとうございました

© XML Consortium