IBM Research

Web2.0時代のセキュリティ

日本IBM 東京基礎研究所
丸山宏・佐藤　史子

XMLコンソーシアムDay
December 12, 2006

アジェンダ

- 第1部（丸山）
  - セキュリティとは何か
  - Web2.0とは何か
  - Web2.0におけるセキュリティのチャレンジ
- 第2部（佐藤）
  - マッシュアップされたサービスのセキュリティ
  - Webサービスセキュリティ関連仕様
  - Webサービスセキュリティ適用における問題

情報セキュリティは、今まで機密性 (C)、完全性(I)、可用性(A)の要件として定義されてきた

- **機密性** (Confidentiality)
  - **許可を得ない人が、情報を読み出してはならない**
- **完全性** (integrity)
  - **許可無く情報が書き換えられない (改ざんされない)**
- **可用性** (Availability)
  - **必要な時にいつも情報にアクセスすることができる**

しかし、それだけではどういう場合にC.I.A.が成立しなければならないのか、破られたときにどのような損害が発生するのか、などの観点が不十分

- **機密性** (Confidentiality)
  - **許可を得ない人が、情報を読み出してはならない**
- **完全性** (integrity)
  - **許可無く情報が書き換えられない (改ざ**
    い)
- **可用性** (Availability)
  - **必要な時にいつも情報にアクセスすることができ**

どういう場合か？(ポリシーは何か？)

CIAが成立しないと、どのような損害が発生するか？

個人情報保護法やSOX法などの法制化により、情報セキュリティは情報コンプライアンスリスクに対するリスク管理、という色彩を帯びてきている

- 法制化の動き
  - 個人情報保護法 (日本)
  - Sabanes & Oxley 法
  - …
- 情報コンプライアンス＝情報の取り扱いに関するルール･ポリシーを守ること
  - 法令：個人情報保護法、SOX法、…
  - 企業のセキュリティ･ポリシー
  - 企業のプライバシー･ポリシー
  - …
- コンプライアンスを守れなかった場合に損害が発生する → リスクがある
  - アーサーアンダーセン社 (エンロン粉飾決算)
  - ソフトバンク社の個人情報漏洩
  - …
- リスクはゼロにはできないが、管理することは可能
  - リスク管理の考え方の導入

リスク管理の考え方からは、情報セキュリティは、「セキュアかどうか」ではなく、「リスクの程度は何か」で考えなければならない

“このシステムはセキュアである”

“このシステムは100%セキュアである”

“このシステムはよりセキュアである”

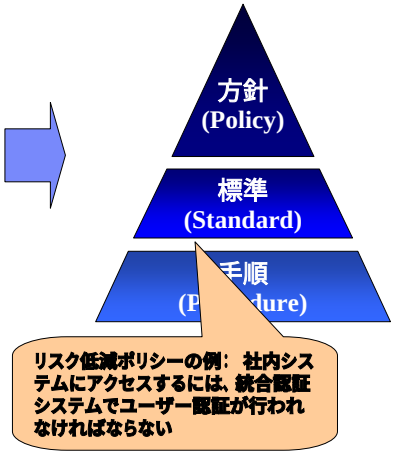
“このシステムのリスクはXXである”

ポリシーとはリスクに対する戦略を具現化したもの

リスクに対する戦略

- リスク回避（リスクのあるビジネスを行わない）
- リスク受容（リスクの受け入れ）
  - 事故があった時の対策を立てておく
- リスク低減
  - 脅威の低減 (e.g., 従業員の教育)
  - 脆弱性の低減 (e.g., 暗号化)
  - 被害の低減（e.g., ファイヤーウォール）
- リスク転嫁
  - アウトソーシング
  - 保険

企業のいわゆる「ポリシー」



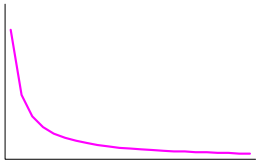
アジェンダ

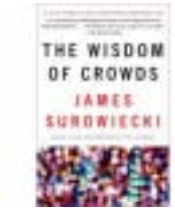
- 第1部（丸山）
  - セキュリティとは何か
  - **Web2.0とは何か**
  - Web2.0におけるセキュリティのチャレンジ
- 第2部（佐藤）
  - マッシュアップされたサービスのセキュリティ
  - Webサービスセキュリティ関連仕様
  - Webサービスセキュリティ適用における問題

Web2.0とは何か

キーワード

- **テクノロジー**
  - Wiki、AJAX、RSS、ATOM、LAMP、
- **サービス**
  - **アフィリエイツ、アドセンス、**
  - **ソーシャルネットワーク** (e.g., Mixi) 、WikiPedia
  - Google Map
- **考え方**
  - **ロングテール**
  - Wisdom of Crowds
  - **マッシュアップ**





J. Surowiecki, “The Wisdom of Crowds” ASIN: 0385721706

本質は, Webが具現化する文化

イノベーションのあり方

- IBM “Global Innovation Outlook”
  - **「マスマーケットではなく、個人個人に対応することによって新たないノベーションが現れる」**
    - Amazon.com
    - **製薬における「Generation You」**
- Von Hippel, “Democratinzing Innovation”
  - **イノベーション自身もロングテール化（新しいアイディアは個人から現れる）**
    - **オープンソース**

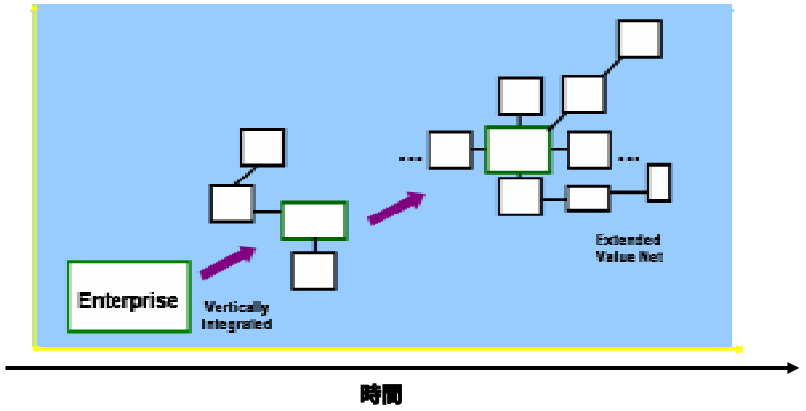
IBM Global Innovation Outlook

Von Hippel: *Democratizing Innovation*

個人の重要性

産業におけるエコシステム

- 昔：部品の生産から製品の出荷、サービスまで一貫して提供
- 今：パートナー、サプライヤ、カスタマー、競争相手などからなる「エコシステム」が産業を支える



未来の企業のあり方

出典：*IBM Global Innovation Outlook 2.0*

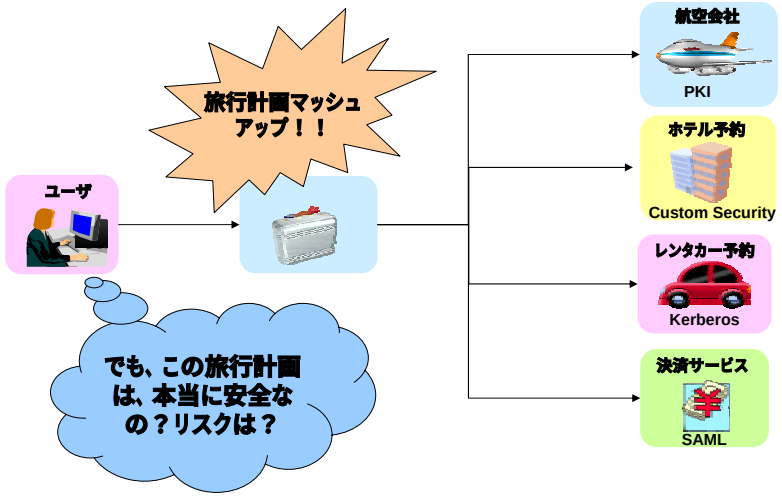
■「企業」の定義そのものが問い直されている
- ■固定的でない、流動的な雇用形態
- ■「人々を結び付けるのは共通の目的『エンデバー』になりつつある」
  - ■ ハリウッドの例。映画監督、脚本家、俳優、カメラ、大道具、コンピュータ・グラフィックス。

Mash Upが21世紀の産業のあり方になる

アジェンダ

- 第1部（丸山）
  - セキュリティとは何か
  - Web2.0とは何か
  - **Web2.0におけるセキュリティのチャレンジ**
- 第2部（佐藤）
  - マッシュアップされたサービスのセキュリティ
  - Webサービスセキュリティ関連仕様
  - Webサービスセキュリティ適用における問題

Web2.0時代のセキュリティ



マッシュアップが信頼できるためには、1) 関係者のポリシーと責任が明確になっていること、 2) その信頼の度合いがわかりやすいこと、の2点が重要

- ポリシーと責任の明確化
  - それぞれの部品サービスは、どのようなポリシーを持っているだろう？
  - この結果はどのようないきさつで出てきたのだろう？

  →すべての計算結果について、ポリシーと責任の所在を明確にできることが必要。このことによって、関係者がより責任を持って事に当たるようになる
- わかりやすさ
  - 自分はこのマッシュアップを使うことによって、自分の生命や財産をどの程度危険にさらすのだろう？

  →常にユーザが判断できるだけのわかりやすさが必要

複雑さは常にセキュリティの最大の敵

- ほとんどの場合、トータルセキュリティの中でのウィークポイントは人。
- 人は常にリスク判断を下している
  - 道を渡るとき (車が来るかもしれない)
  - パスワードを入力するとき (誰かが見ているかもしれない)
  - Webサーフィンをしているとき (悪いプログラムが入ってくるかもしれない)
  - :
- 状況が複雑になると、正しい判断を下しにくい
  - インターネットキオスクでWebサーフィン中に見つけたショッピングサイト。これを使っても安全か？
    - 端末：OSのパッチは最新か、AntiVirusの定義ファイルは最新か、ファイアウォールは正しく設定されているか、スパイウェアは入っていないか…
    - サイト：サイトは本物か、通信は暗号化されているか、脆弱性のあるソフトを使っていないか？
    - アプリ：アプリのその他のコンポーネントは十分に信頼できるか？
    - 運営者：運営者は信頼できるか？
    - 保険：問題が起きたときに保障があるか？
    - …

リスク判断を行うには、より単純化されたセキュリティモデルが役に立つはず

単純化されたセキュリティクラス (例)

- Class A
  - すべてのコンポーネントが政府レベルの認証をパスしています
  - すべての関係者が政府レベルの認証をパスしています
  - 問題が起きた場合にすべての損害が補償されます
- Class B
  - すべてのコンポーネントが商用グレードの認証を持っています
  - すべての関係者の身元が特定されています
  - 問題が起きた時の責任の所在が明確です
- Class C
  - すべての関係者の身元が特定されています
- Class D
  - 何もわかりません

第1部のまとめ

1. セキュリティとはリスク管理である
2. Web2.0の世界は、ソーシャルネットワークの世界である
3. Web2.0におけるセキュリティのチャレンジは、
   1. ポリシーと責任の明確化
   2. 素人にもわかりやすいセキュリティモデル

IBM Research

Web2.0時代のセキュリティ
- Webサービスセキュリティの最新動向 -

日本IBM 東京基礎研究所
佐藤　史子

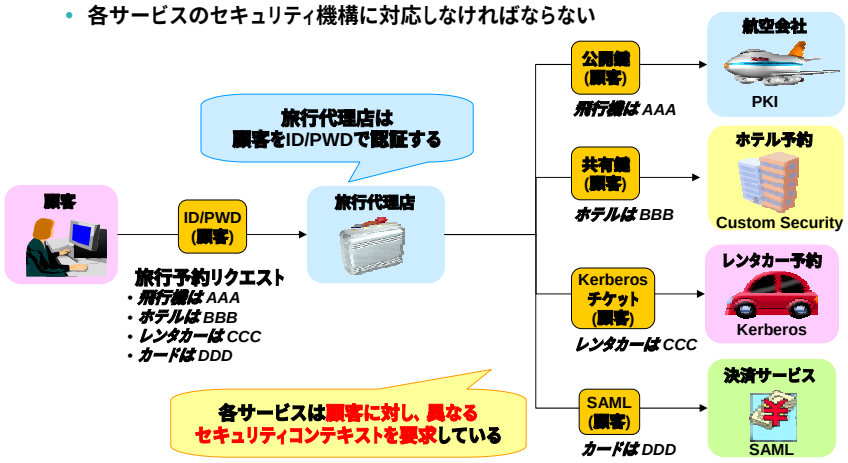
XMLコンソーシアムDay
December 12, 2006

第2部アジェンダ

- マッシュアップされたサービスのセキュリティ
- Webサービスセキュリティ関連仕様
  - 標準化状況
  - WS-Security 関連仕様
    - WS-Trust
    - WS-SecureConversation
    - WS-SecurityPolicy
- Webサービスセキュリティ適用における問題
  - モデル駆動型セキュリティ
  - セキュリティポリシー自動生成
- まとめ

マッシュアップされたサービスのセキュリティ

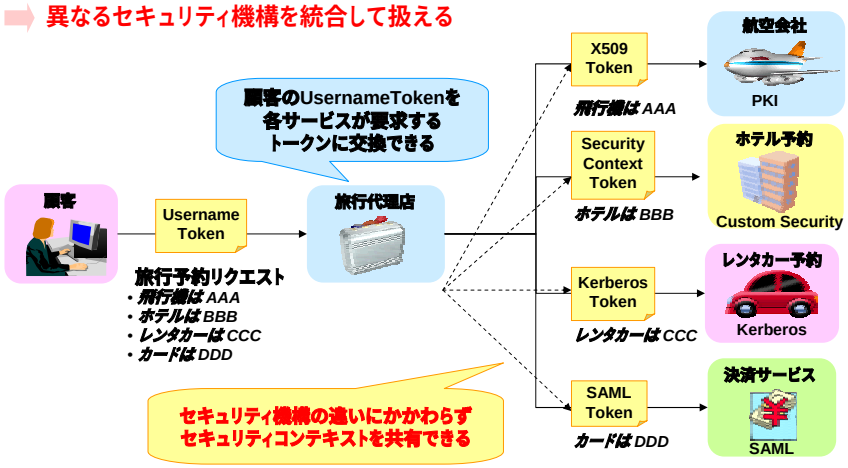
- 旅行代理店は顧客のリクエストを各サービスに転送する
  - 各サービスに対し、顧客のセキュリティコンテキストを転送する
- サービスはマッシュアップされても、各サービスのセキュリティ機構は異なったまま
  - 各サービスのセキュリティ機構に対応しなければならない



Webサービスセキュリティを適用すると…

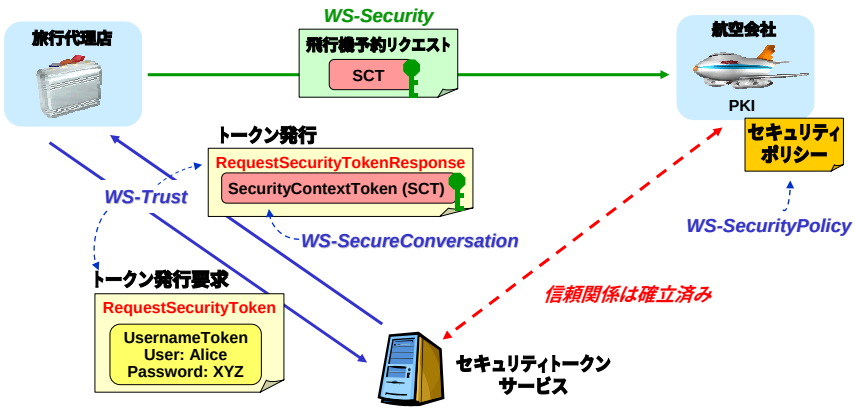
- セキュリティコンテキストを抽象化してセキュリティトークンとして表現する
- 異なるセキュリティトークンは交換できる
  - 旅行代理店は顧客のトークンを変換して各サービスに転送できる

➡ 異なるセキュリティ機構を統合して扱える



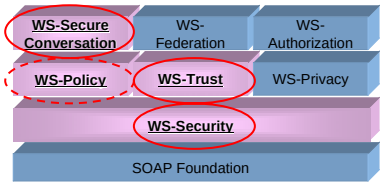
セキュリティトークンの交換

- セキュリティトークンはセキュリティトークンサービスにより発行・交換できる
  - 各サービスとセキュリティトークンサービスの信頼関係はあらかじめ確立されている
- 発行されたセキュリティトークンを使うことで、安全なセッションを確立できる
  - 共通のセキュリティトークンによる署名・暗号化が可能になる



Webサービスセキュリティ関連仕様

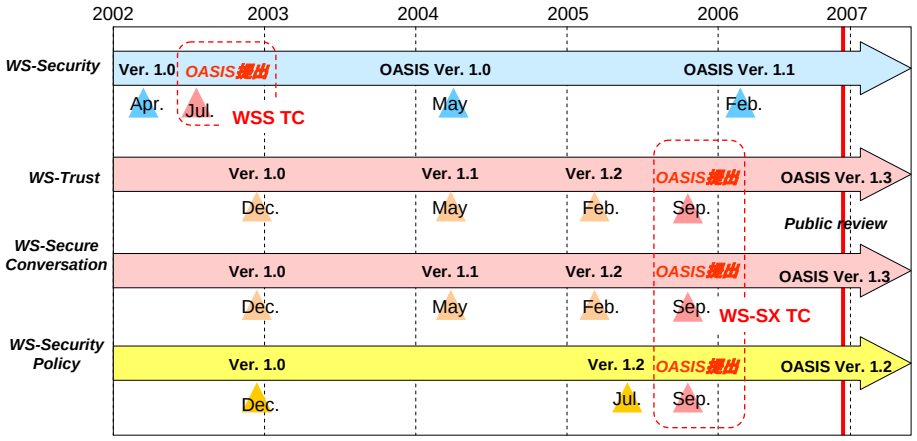
- Web Services Security
  - Webサービスセキュリティの基本となるビルディング・ブロック
  - SOAPヘッダに付くSecurityヘッダの基本構造を定義
    - デジタル署名
    - 暗号化
    - セキュリティトークン
  - 複数のセキュリティトークンプロファイル
    - Username Token Profile
    - X509 Token Profile
    - SAML Token Profile など



- Web Services Trust
  - 異なるセキュリティドメイン間で信頼を確立するための方法を定義
- Web Services Secure Conversation
  - セキュアなセッションで共有するセキュリティコンテキストトークンを定義
- Web Services Security Policy
  - WS-Policy に基づいてセキュリティポリシーを定義
    - WS-Policy そのものは別に標準化されている

標準化状況

- OASISで標準化活動中
  - WSS TC: WS-Security Core, Token Profiles
  - WS-SX TC: WS-Trust, WS-SecureConversation, WS-SecurityPolicy



WS-Security

- SOAPメッセージを署名・暗号化する
  - 鍵はセキュリティトークンにより表現
- セキュリティトークンの種類
  - UsernameToken
    - ID/Password の表現
  - X509Token
    - X509証明書の表現
    - 公開鍵をエンコードした BinarySecurityToken
  - SAMLToken
    - Security Assertion Markup Language (SAML) の表現
    - 対象鍵が含まれる
  - KerberosToken
    - Kerberos Ticket の表現
- カスタムなセキュリティトークンも定義できる

```
<soapenv:Envelope>
 <soapenv:Header>
  <wsse:Security soapenv:mustUnderstand="1"
    xmlns:wsse="http://docs.oasis-
  open.org/wss/2004/01/oasis-200401....">
    <wsse:BinarySecurityToken
       ValueType="...#X509v3"
       EncodingType="...#Base64Binary>
          MIIEZzCAwIBAgIQEmtJZc0...
    </wsse:BinarySecurityToken>
    <ds:Signature>
      <ds:SignatureValue>
        SbhHeGPrsoyxXbTPdIEcybfqvoEdZ/dvg...
      </ds:SignatureValue>
    </ds:Signature>
  </wsse:Security>
 </soapenv:Header>
 <soapenv:Body>
  <EncryptedData....>
    <EncryptionMethod
  Algorithm="http://....001/04/xmlenc#tripledes-cbc"/>
      .....
  </EncryptedData>
 </soapenv:Body>
</soapenv:Envelope>
```

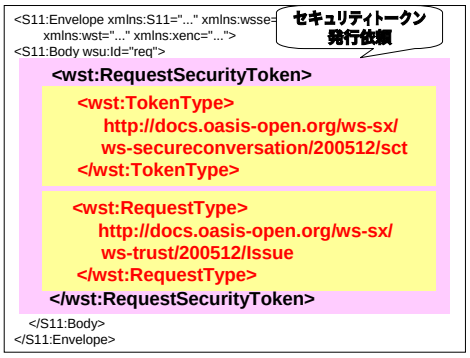
セキュリティトークン

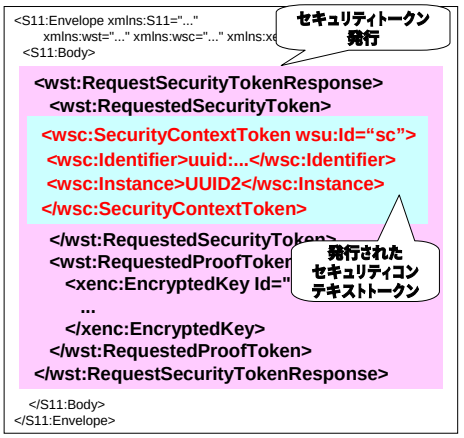
署名

暗号化された部分

WS-Trust & WS-SecureConversation

- WS-Trust
  - セキュリティコンテキストを交換し信頼を確立するためのメッセージを定義
- WS-SecureConversation
  - 信頼が確立されたセッションで共有するセキュリティコンテキストトークンを定義

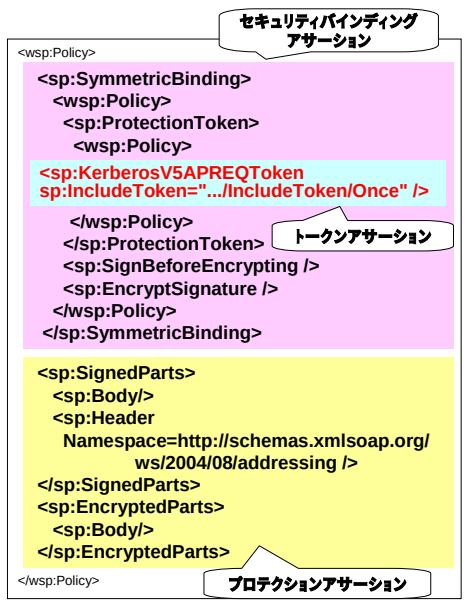




WS-SecurityPolicy

- セキュリティポリシー
  - Webサービスに要求されるセキュリティ要件の記述
- 複数のアサーションによりポリシーを記述
  - Security Binding Assertion
    - 署名・暗号化の方法
  - Protection Assertion
    - 署名・暗号化の対象
  - Token Assertion
    - セキュリティトークンの種類

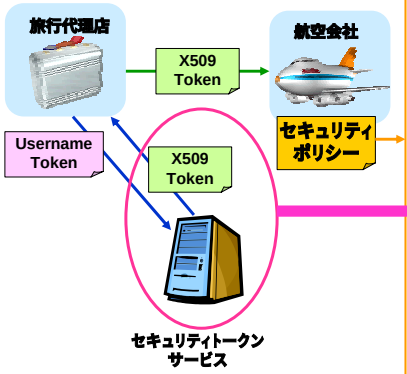
- Token Assertion
  - UsernameToken
  - X509TokenAssertion
  - KerberosTokenAssertion
  - IssuedTokenAssertion



Webサービスセキュリティ適用における問題

- セキュリティポリシーを正しく記述するにはさまざまな知識・理解が必要
  - Webサービスセキュリティ関連仕様
  - アプリケーションサーバやセキュリティトークンサービスの設定

➡セキュリティポリシーの記述は非常に複雑で難しい



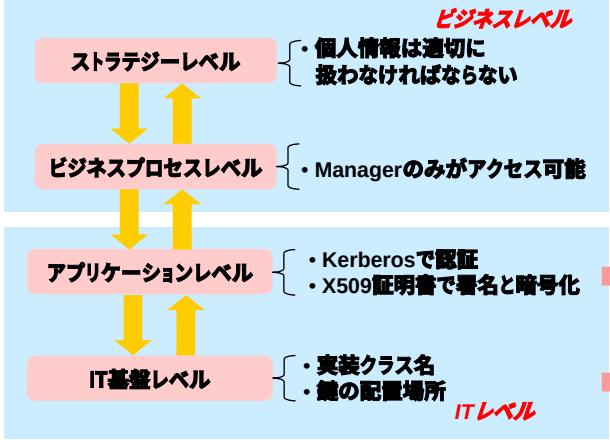

```
<sp:SupportingTokens>
 <wsp:Policy>
  <sp:IssuedToken…>
   <sp:Issuer>
   <wsa:Address>http://www.sts.issue</wsa:Address>
   </sp:Issuer>
   <sp:RequestSecurityTokenTemplate …>
    <wst:RequestSecurityToken …>
  <wst:TokenType>http://…#X509v3</wst:TokenType>
      <wst:RequestType>
       http://.../trust/Issue
      </wst:RequestType>
    </wst:RequestSecurityToken>
   </sp:RequestSecurityTokenTemplate>
   <sp:RequireDerivedKeys />
   <sp:RequireExternalReference />
  </sp:IssuedToken>
</wsp:Policy>
</sp:SupportingTokens>
```

プラットフォームの設定に依存

WS-Trustなど仕様の理解が必要

Webサービスセキュリティポリシー

- 一般的にセキュリティポリシーとは、ストラテジーレベル または ビジネスプロセスレベルのポリシーをさすことが多い
- Webサービスのセキュリティポリシーは、ITレベルのセキュリティポリシーに対応



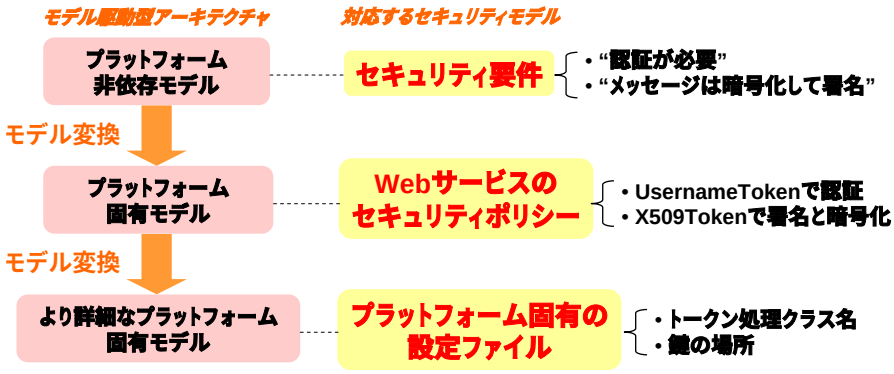
IT基盤（プラットフォーム）に依存した情報を記述

Webサービスのセキュリティポリシー

プラットフォーム固有の設定ファイル

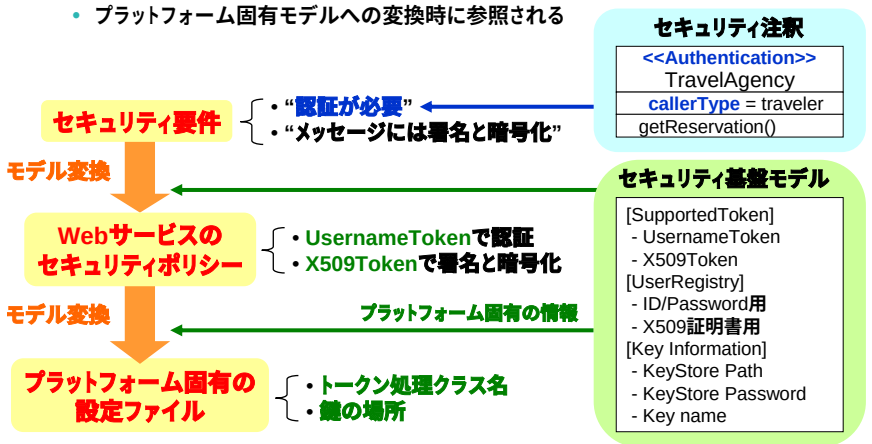
モデル駆動型アーキテクチャの適用

- セキュリティポリシーは、プラットフォームのセキュリティ機能に依存する
  - ➡ **プラットフォーム固有モデルであるとみなせる**
- **モデル駆動型アーキテクチャを適用して、セキュリティポリシーを自動生成する**
  - プラットフォーム非依存モデルを導入し、抽象的なセキュリティ要件を表現する



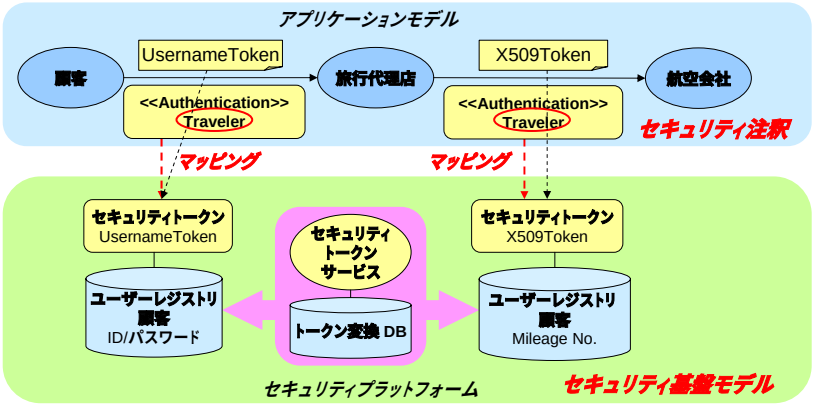
モデル駆動型セキュリティ

- プラットフォーム非依存モデルを定義: セキュリティ注釈
  - セキュリティの要件を**抽象的な注釈**で表す
- プラットフォーム固有の情報をモデル化: セキュリティ基盤モデル
  - プラットフォーム固有モデルへの変換時に参照される



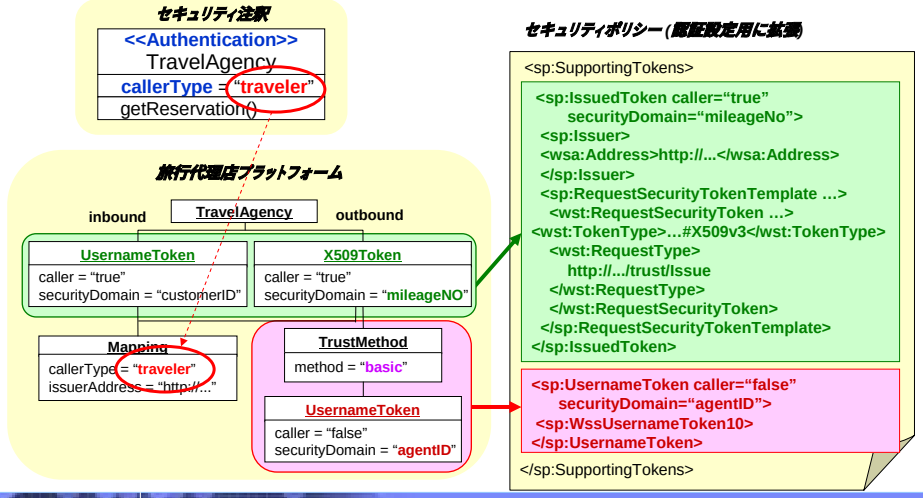
セキュリティ注釈とセキュリティ基盤モデルのマッピング

- 各サービスはそれぞれ異なるセキュリティトークンを要求する
  - セキュリティトークンの形式は異なるが、本質的には同等
- セキュリティ注釈により、顧客認証そのものを設定できる
  - セキュリティトークンの種類などプラットフォームの情報は考慮する必要はない
  - 注釈をセキュリティ基盤モデルにマッピングすることで、ポリシーが自動生成できる



セキュリティポリシー自動生成

- マッピングされたプラットフォームの情報を参照して自動生成する
- プラットフォームに依存しないルールを定義でき、ポリシーの自動生成が可能になる



まとめ

- Webサービスセキュリティは既存の異なるセキュリティ機構を統合する
  - セキュリティトークンという抽象表現の導入により、異なるセキュリティコンテキストの交換が可能になる
- セキュリティトークン交換のための関連仕様がOASISで標準化活動中
  - WS-Security Ver.1.1
  - WS-Trust Ver.1.3
  - WS-SecureConversation Ver.1.3
  - WS-SecurityPolicy Ver.1.2
- 複雑なセキュリティポリシーを自動生成する手法を提案
  - 複数のセキュリティ機構を統合した環境では、セキュリティポリシーの記述は非常に複雑になる
  - モデル駆動型アーキテクチャを適用したセキュリティポリシーの自動生成を研究中