# セキュリティ関連XML規格の動向

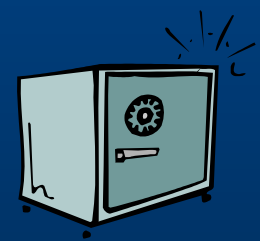
2003年6月2日

XML Consortium 応用技術部会

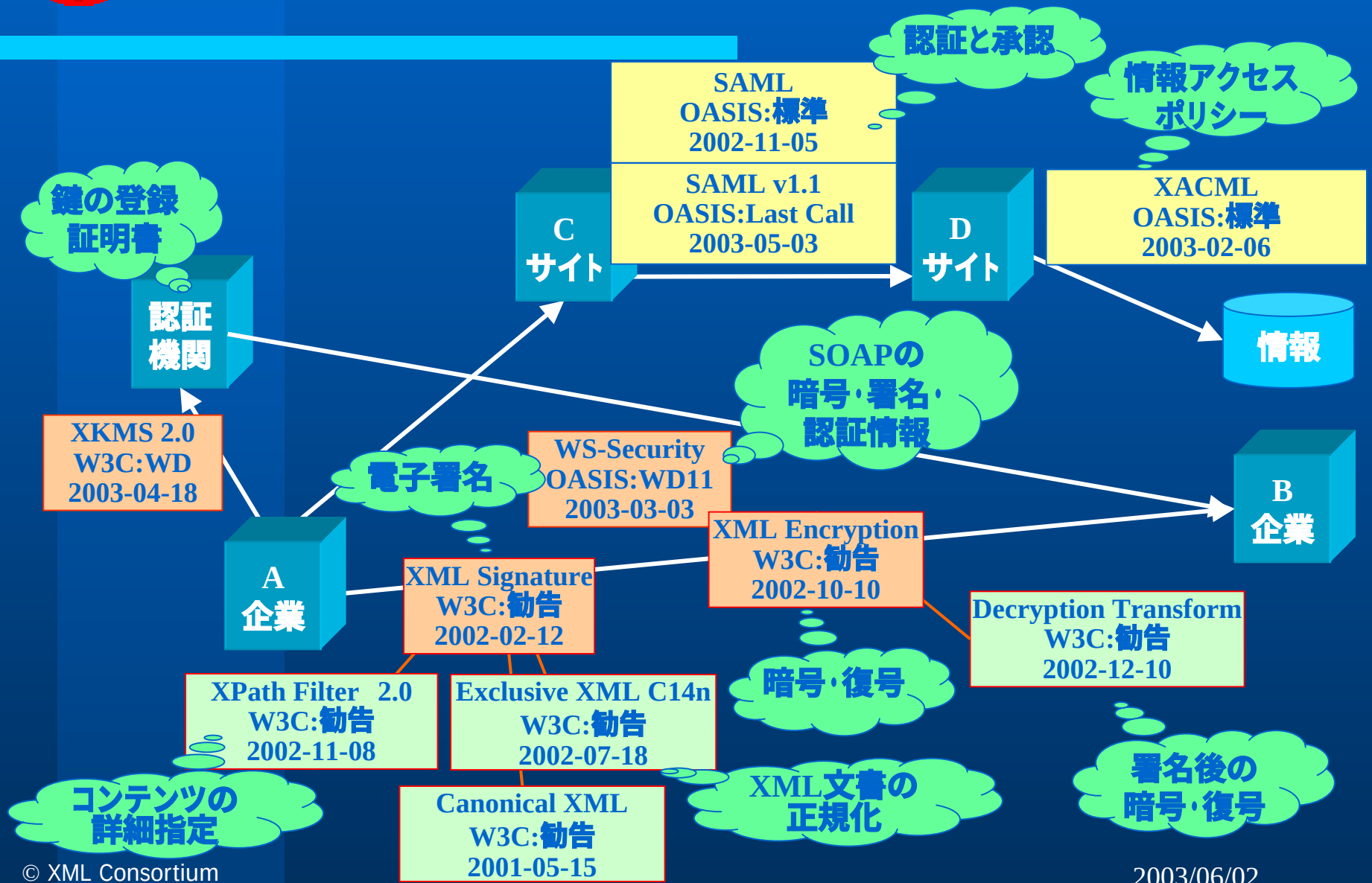
ミノルタ株式会社　上田隆司

# 本日の説明内容

- セキュリティ関連XML規格の動向
  - 電子署名：XML Signature
  - 暗号化：XML Encryption
  - 公開鍵基盤：XKMS
  - Webサービスセキュリティ：WS-Security
  - シングルサインオン：SAML
  - アクセス制御：XACML
  - まとめ



# セキュリティ関連XML規格一覧



# XML Signature

- **XML Signature**

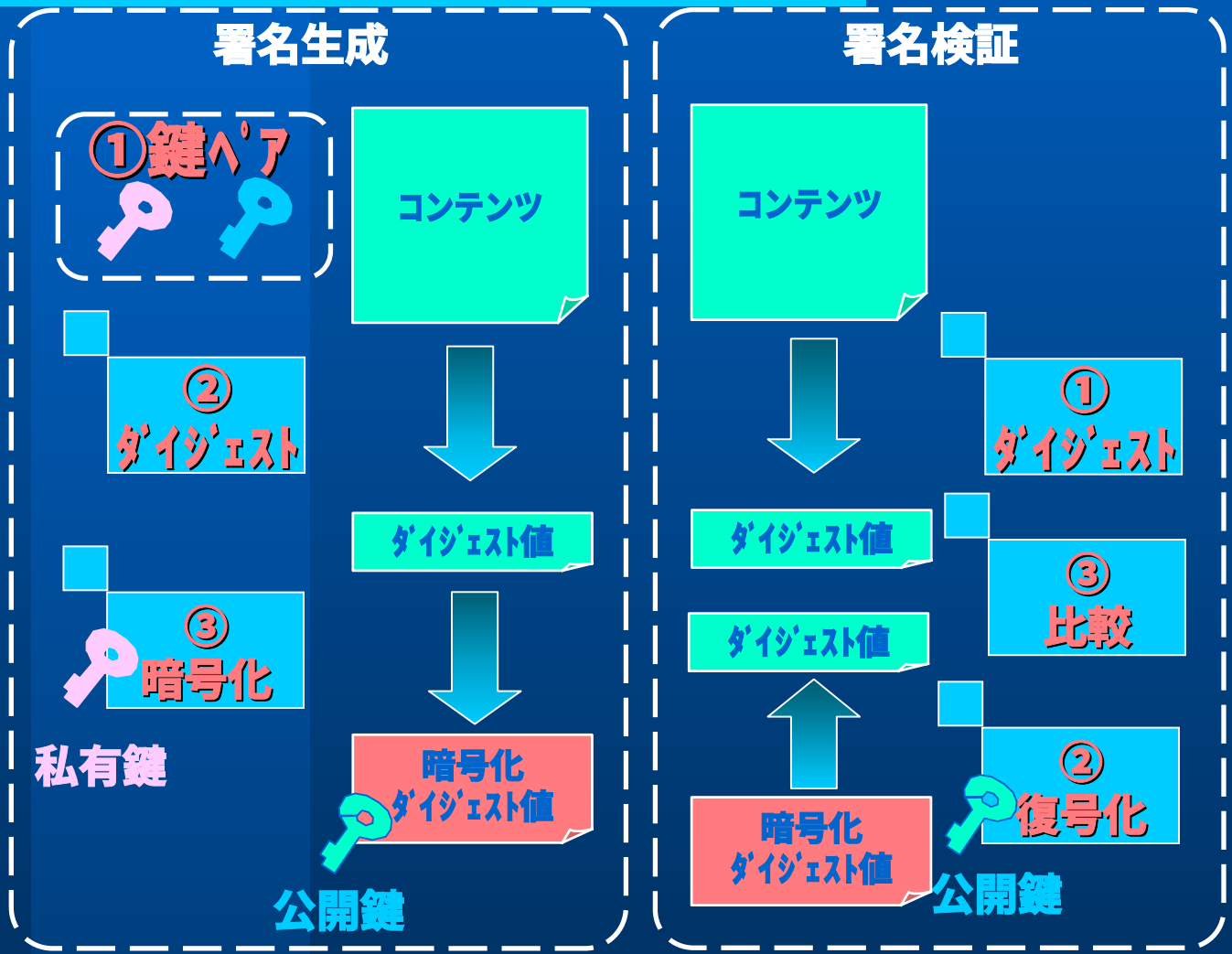
  **http://www.w3.org/Signature/**

- **セキュリティ上の目標**
  - **コンテンツの改ざんを検出する**
  - **PKIと連携し送信者の否認を防止する**
- **XML Signatureが決めるもの**
  - **署名検証に必要な情報（アルゴリズム、鍵情報）のXML表現とその処理手順**

# XML Signature

- 経過
  - W3C:XML Signature WG
    - 1999年9月 Requirement完了
    - IETFと共同作業
    - 2002年2月12日 W3C勧告（http://www.w3.org/TR/xmldsig-core/）
    - 2002年3月 IETF Working Draft（RFC3275）
- 特徴
  - PKCS#7をより柔軟に
    - 署名対象の指定
      （複数コンテンツ、外部コンテンツ、部分に署名）
    - 署名形式（Detached、Enveloped、Enveloping）
    - 鍵表現

# 署名処理手順（公開鍵暗号）




署名対象
(コンテンツ)

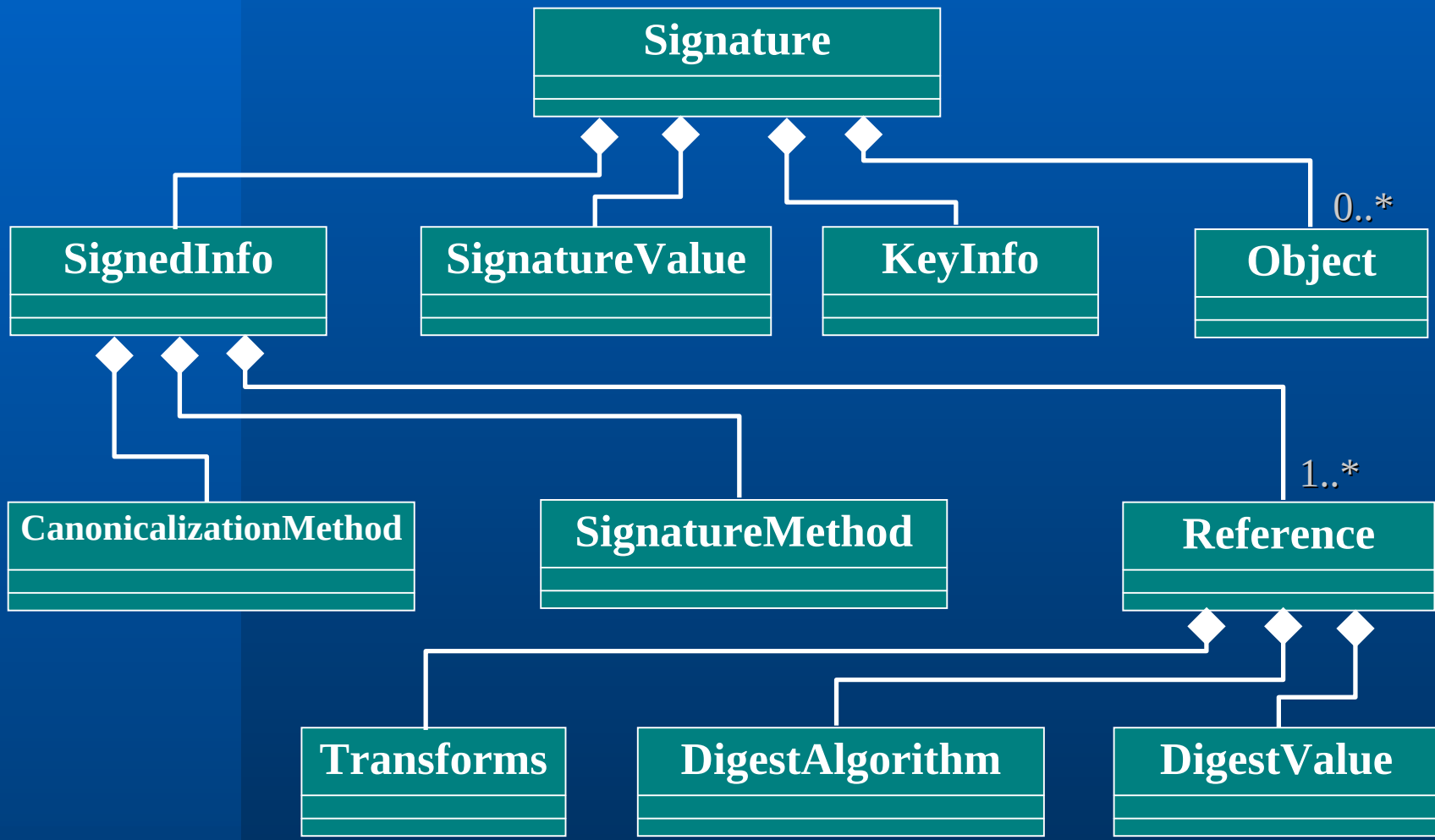
ﾀﾞｲｼﾞｪｽﾄ
ｱﾙｺﾞﾘｽﾞﾑ

暗号・復号
ｱﾙｺﾞﾘｽﾞﾑ

公開鍵

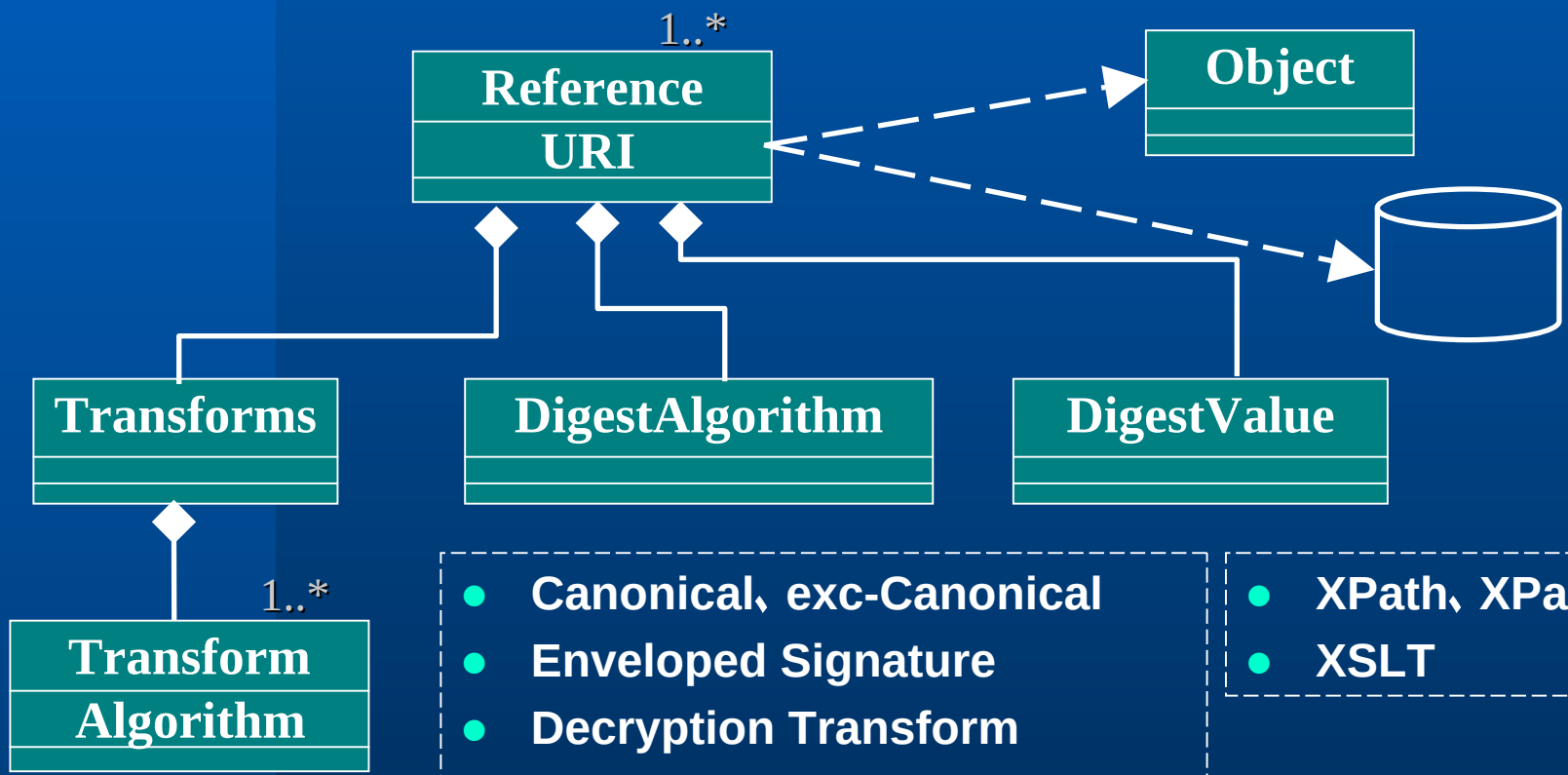
暗号化
ﾀﾞｲｼﾞｪｽﾄ値

# メッセージ構成



# 署名対象の指定

- <Reference>でコンテンツを間接指定
- <Transform>で部分・詳細・変換を指定可能




- Canonical、exc-Canonical
- Enveloped Signature
- Decryption Transform
- Base64

- XPath、XPath 2.0
- XSLT

# 署名形式

| Enveloping | Enveloped | Detached |
|---|---|---|
| <ds:Signature><br><ds:Object><br><myap:Order/><br></ds:Object><br></ds:Signature > | <myap:Order><br><ds:Signature><br></ds:Signature ><br></myap:Order> | <myap:Order><br></myap:Order><br><ds:Signature><br></ds:Signature > |
| ●コンテンツと署名を一括管理<br>●署名処理が前提 | ●コンテンツと署名を一括管理<br>●<Signature>を理解できる処理系のみ検証 | ●コンテンツと署名を分けて管理<br>●署名ファイルを処理することで検証 |

# 構文例

```
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod
   Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod  Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
     <Reference URI="#Ref1">
      <Transforms>
       <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
     </Reference>
  </SignedInfo>
  <SignatureValue>MC0CFFrVLtRlk.....=</SignatureValue>
  <KeyInfo>
    <KeyName>shimoda@o-camera.com#RSAKey</KeyName>
  </KeyInfo>
  <Object Id="Ref1">
    <myap:Order xmlns:myap="http:xmlcon.com">
      <myap:企画名>空で行くXコン阿波踊りと食いだおれ</myap:企画名>
       <myap:Creditcard>
         <myap:Name>Takashi Shimoda</myap:Name>
         <myap:VALIDTHRU>03-05</myap:VALIDTHRU>
         <myap:CardNo>1234-5678-9999-0000</myap:CardNo>
       </myap:Creditcard>
    </myap:Order>
  </Object>
</Signature >
```

# その他

- Interoperability
  - http://www.w3.org/Siganture/2001/04/05-xmldsig-interop.html (16)
- Availability
  - パッケージレベルも含め、多数の製品あり
    - Source:5
    - Toolkit:12
- その他
  - Java API標準の策定

# 関連規格

- Canonical XML

  http://www.w3.org/TR/xml-c14n

- セキュリティ上の目標
  - 処理系による差異を吸収
    - 文字コード
    - 終了タグ処理. . . . .

<CardType vender=‘Master’></CardType>

<CardType____vender=“Master”/>

意味的には同じだが、
ダイジェストが異なる

# 関連規格 (続)

- 経過
  - W3C:XML Signature WG
    - 2001年3月15日 勧告
      (http://www.w3.org/TR/xml-c14n)
- 処理概要
  - UTF-8に変換
  - ラインブレーク→#xA
  - 終了タグを追加

–空白の処理
–エンティティの復元
–etc.

<CardType vender='Master'></CardType>

<CardType____vender="Master"/>

<CardType vender="Master"></CardType>

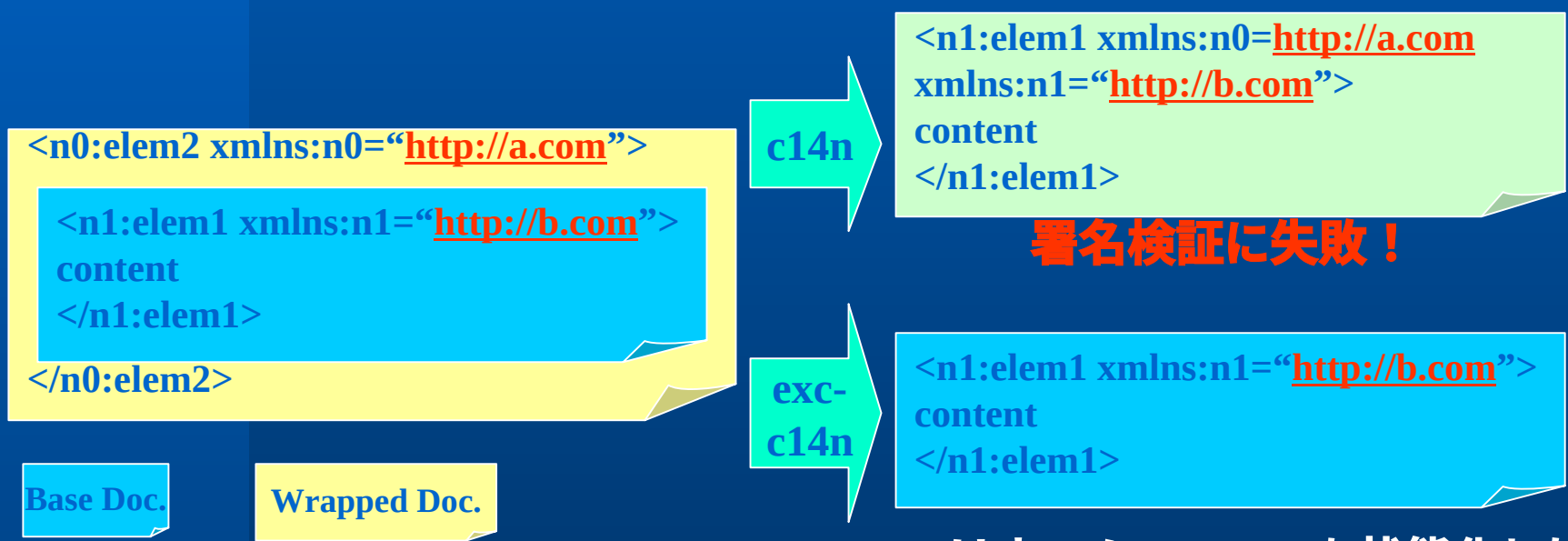
# 関連規格（続）

- Exclusive XML Canonicalization
  http://www.w3.org/Signature/

- 規格の目的
  - C14Nのネームスペース問題を解決
    （別文書でラッピング問題）
- 経過
  - W3C:XML Signature WG
    - 2002年7月18日 勧告
      （http://www.w3.org/TR/xml-c14n）

# 関連規格 (続)

- 特徴
  - ネームスペースの状態化を排他的に制限



# 関連規格（続）

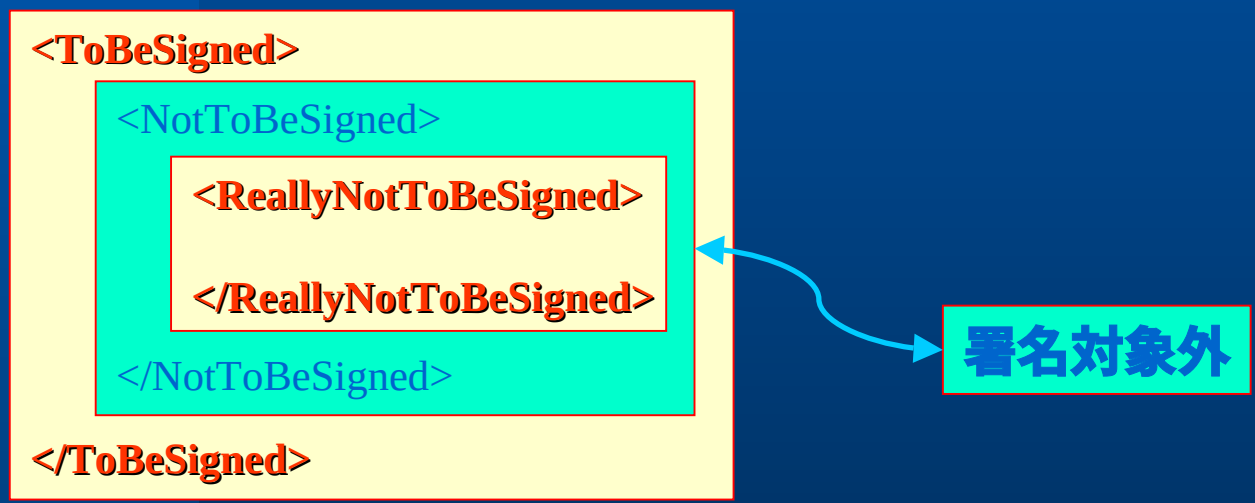
- XML-Signature XPath Filter 2.0
  http://www.w3.org/Signature/

- 規格の目的
  - 排他的な署名対象の指定を実現
    （複数のEnveloped Signatureを実現）
- 経過
  - W3C:XML Signature WG
    - 2002年11月8日 勧告
      （http://www.w3.org/TR/xmldsig-filter2/）

# 関連規格 (続)

- 特徴
  - XPathに横断・除外・結合の表現を追加

```
<XPath Filter="intersect">//ToBeSigned</XPath Filter>
<XPath Filter="subtract">//NotToBeSigned</XPath Filter>
<XPath Filter="union">//ReallyToBeSigned</XPath Filter>
```



# 関連規格 (続)

- Decryption Transform for XML Signature
  http://www.w3.org/Encryption/2001/

- 規格の目的
  - 署名後に暗号化する部分の指定と処理手順
- 経過
  - W3C：XML Encryption WG
    - 2002年12月10日 勧告
      (http://www.w3.org/TR/xmlenc-decrypt)

# 関連規格 (続)

- 特徴
  - <ds:Transform>に復号対象を明示
  - 通信経路での一時的な情報隠蔽に有効

```
<ds:Signature>

  <ds:Reference URI="#Ref1">
    <ds:Transforms>
      <ds:Transform Algorithm="http://www.w3.org/2002/07/decrypt#XML">
        <dcrypt:Except URI="#secret-1"/>
      </ds:Transform>
    </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue> j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
  </ds:Reference>

</ds:Signature>
```

# XML Encryption

- XML Encryption

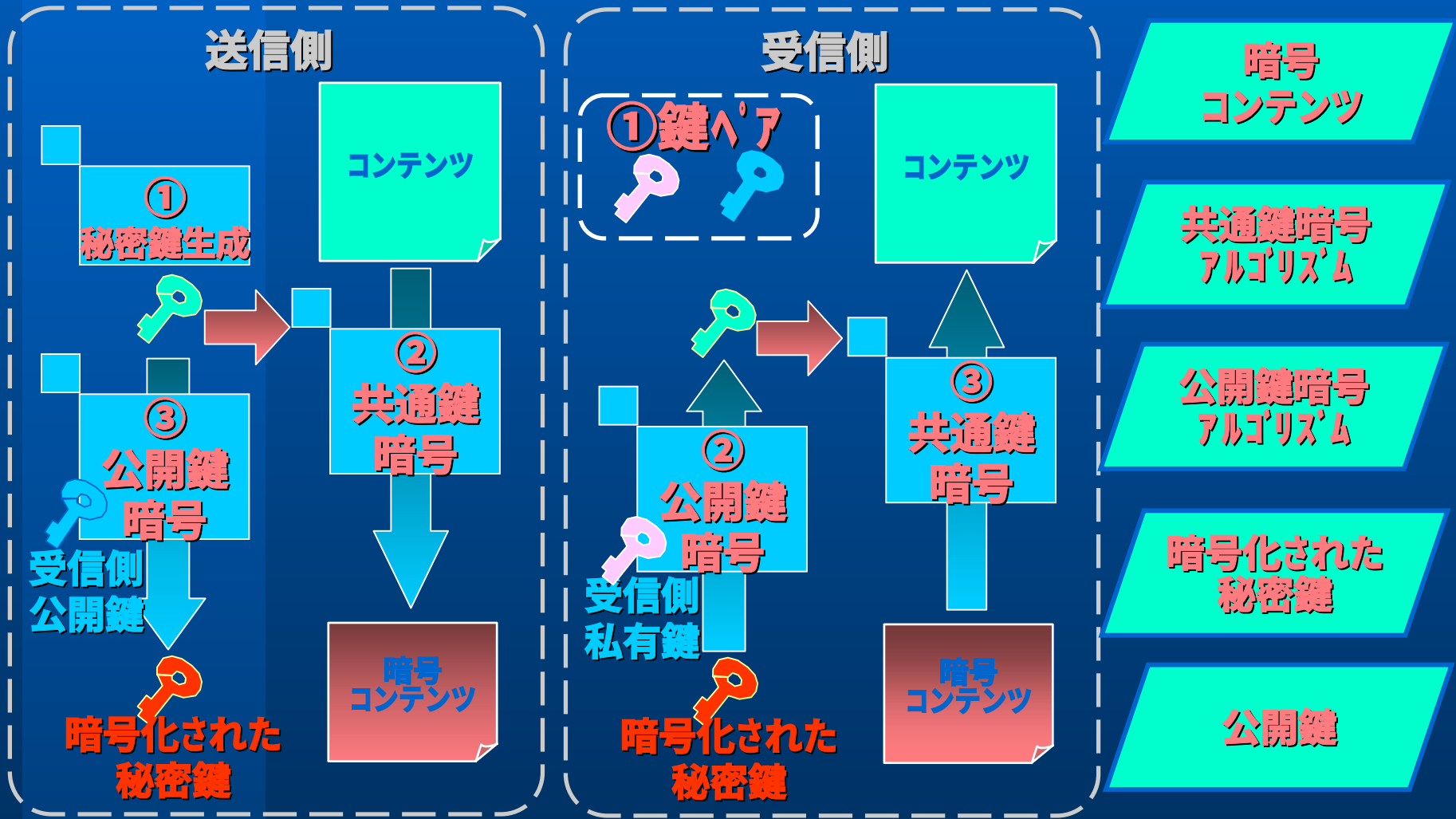
  http://www.w3.org/Encryption/2001/

- セキュリティ上の目標
  - コンテンツの秘匿性を確保
  - (改ざんの防止)
- XML Encryptionが決めるもの
  - 復号に必用な情報(アルゴリズム、鍵情報)のXMLによる表現とその処理手順

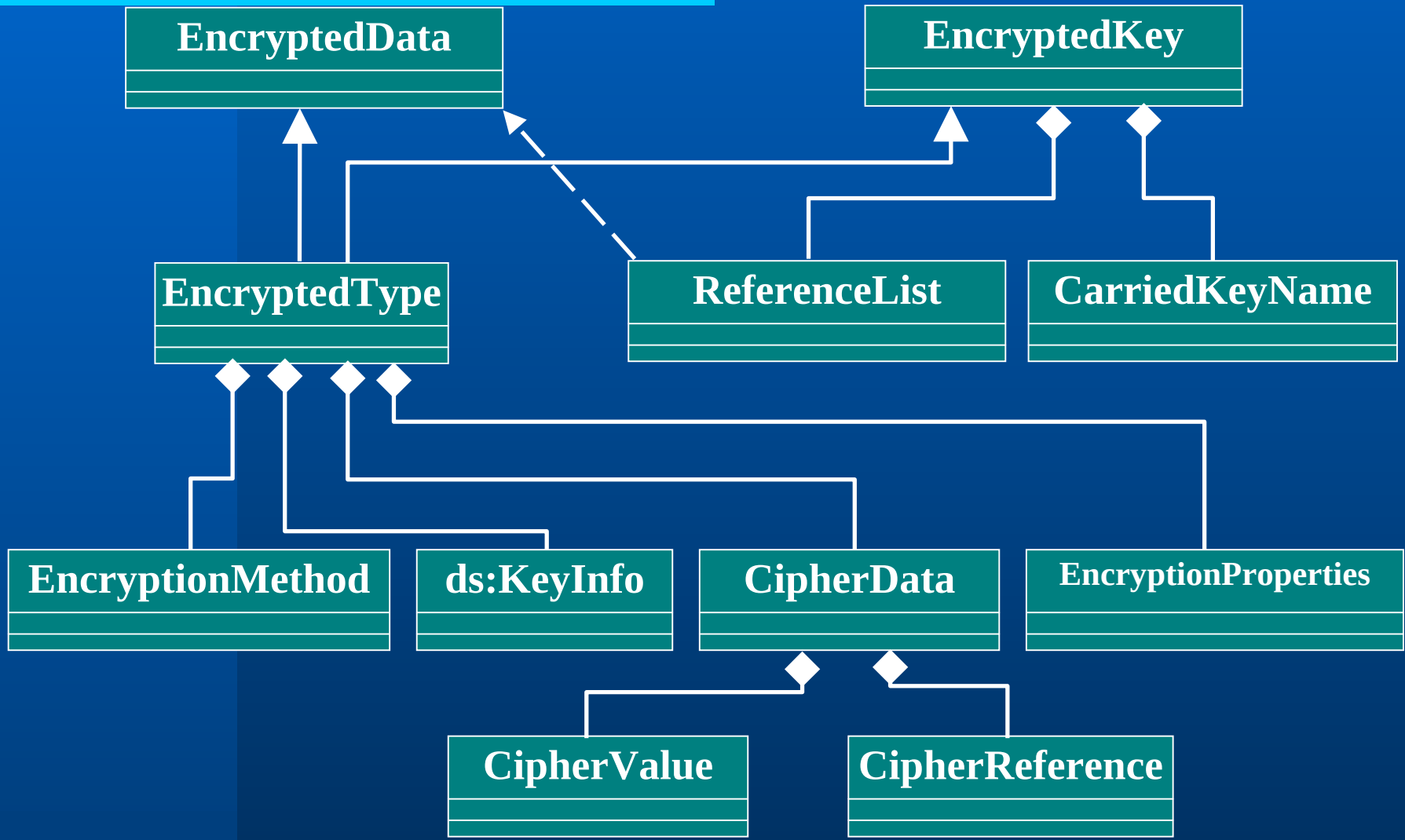
# XML Encryption

- 経過
  - W3C：XML Encryption WG
    - 2001年3月から活動
    - 2002年12月10日 勧告
      (http://www.w3.org/TR/xmlenc-core/)

# 暗号化処理手順（共通鍵の典型）



# メッセージ構成



# 構文例

```
<myap:Order xmlns:myap="http:xmlcon.com" >
 .....
 <myap:Creditcard>
   <xed:EncryptedData Id="ED" xmlns:xed="http://www.w3.org/2001/04/xmlenc#">
        <xed:EncryptionMethod Algorithm="#tripledes-cbc"/>
          <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <ds:RetrivalMethod URI="#EK"/>
          </ds:KeyInfo>
        <xed:CipherData>
          <xed:CipherValue>41a2BdeaXEdda468Xaegde.....</xed:CipherValue>
        </xed:CipherData>
   </xed:EncryptedData>
 </myap:Creditcard>
  .....
 <xek:EncryptedKey Id="EK" xmlns:xek="http://www.w3.org/2001/xmlenc#EncryptedKey">
      <xek:EncryptionMethod Algorithm="#rsa1_5"/>
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
          <ds:KeyName> shimoda@o-camera.com#RSAKey</ds:KeyName>
      </ds:KeyInfo>
      <xek:CipherData>
        <xek:CipherValue>5+GpVuQNTAT3uY8pPed</xek:CipherValue>
      </xek:CipherData>
      <xek:ReferenceList>
        <xek:DataReference URI="#ED"/>
      </xek:ReferenceList>
  </xek:EncryptedKey>
</myap:Order>
```

共通鍵暗号アルゴリズム

暗号コンテンツ

公開鍵暗号アルゴリズム

公開鍵

暗号化された秘密鍵

# その他

- Interoperability
  - http://www.w3.org/Encryption/2002/02-xenc-interop.html (4)
- Availability
  - http://www.w3.org/Encryption/2001/
    - Source:1 (MIT)
    - Toolkit:4 (Baltimore,IBM,Phaos,VeriSign)

# PKI (Public Key Infrastructure)

- セキュリティ上の目標
  - 公開鍵を配布する情報基盤

# 従来技術の問題点

- 鍵登録・失効・回復：
  →プロトコル規定が無い？
  （SCEP：Simple Certificate Enrollment Protocol）
- 暗号化（鍵の取得）：
  →送信先が鍵を登録した第三者機関の特定
  →複数の第三者機関と接続？
- 署名検証（鍵の検証）：
  →公開鍵の検証をクライアントで処理
  - 証明書のパース（ASN.1の解析）
  - 証明書チェーンの解析
  - 第三者機関からのCRL取得と、失効確認

  （OCSPv2、DPV：Delegated Path Validation/ DPD：Delegated Path Discovery/ ）

# XKMS 2.0

- XML Key Management Specification 2.0
  - **http://www.w3.org/TR/xkms2/**

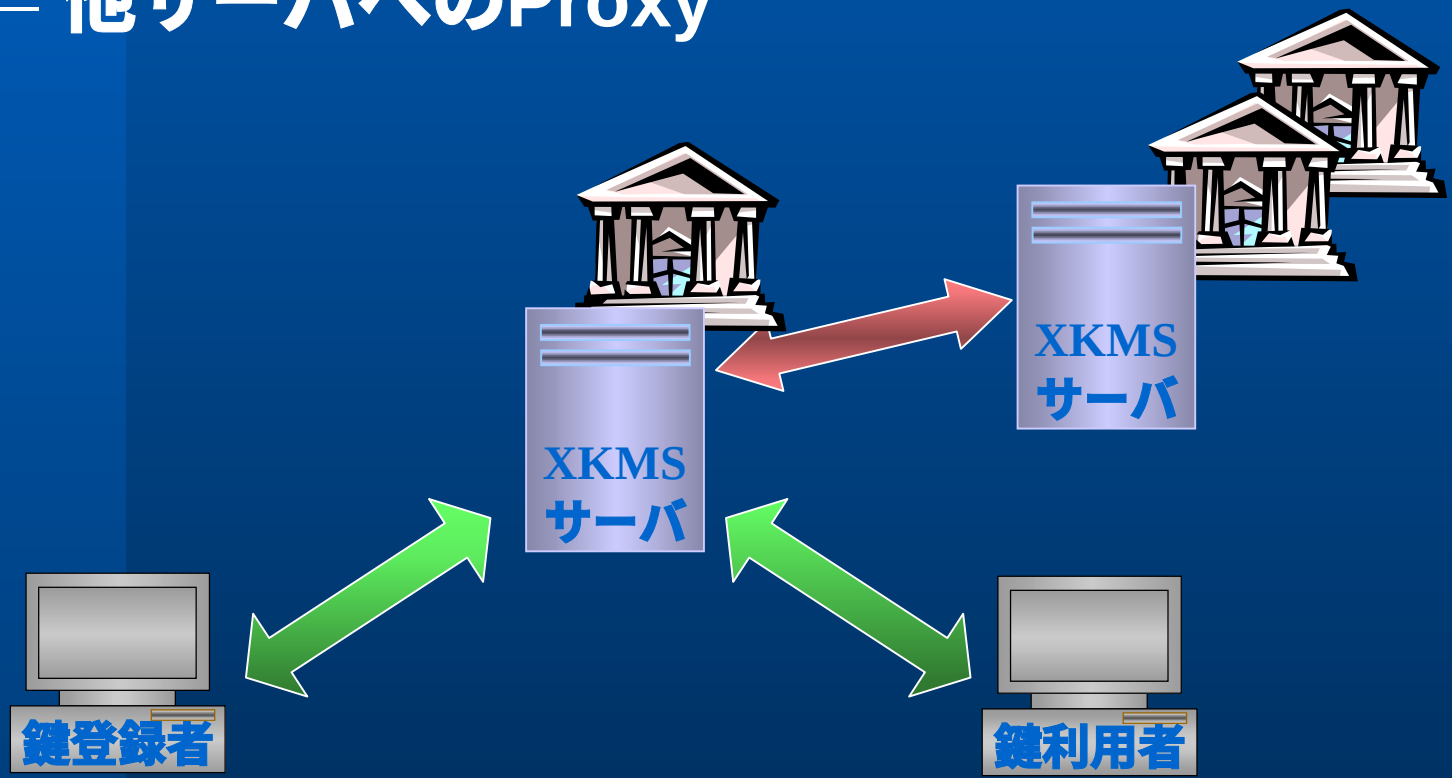
- **XKMSの目標**
  - **既存のPKIを補完する**
    - **PKIを自動化**
    - **鍵利用者の負担を軽減**

# XKMS 2.0 (続)

- 経過
  - OASIS:XML-Based Security Services TC
    - 2000年末から活動
    - 2001年4月にW3CへNote提出(XKMS 1.X)
    - メンバ:VeriSign,Microsoft,WebMethod
  - W3C:XML Key Management WG
    - 2001年12月より活動を開始
    - 2003年04月18日 Working Draft
- 特徴
  - 3サービスを定義
    - K-KRSS (Registry)、X-KISS (Locate、Validate)
  - Webサービスとの高い親和性
  - 鍵情報はXML Signature形式 (<ds:KeyInfo>)

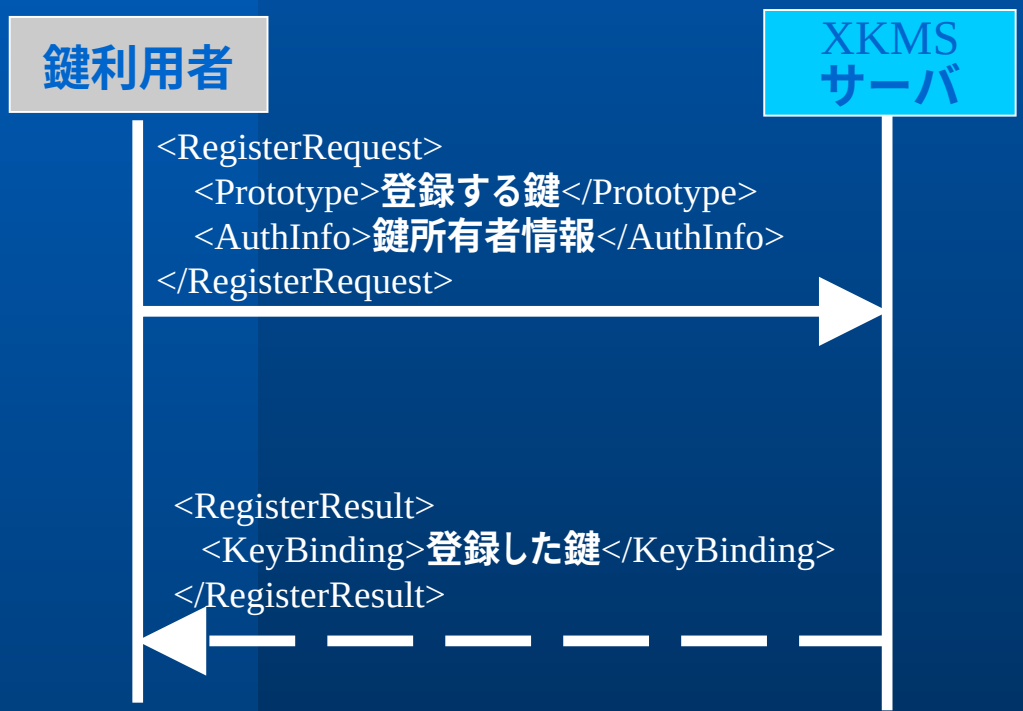
# システム構成

- 既存の第三者機関を補完する
  - 既存の第三者機関のフロントエンド
  - 他サーバへのProxy



# X-KRSS:Register

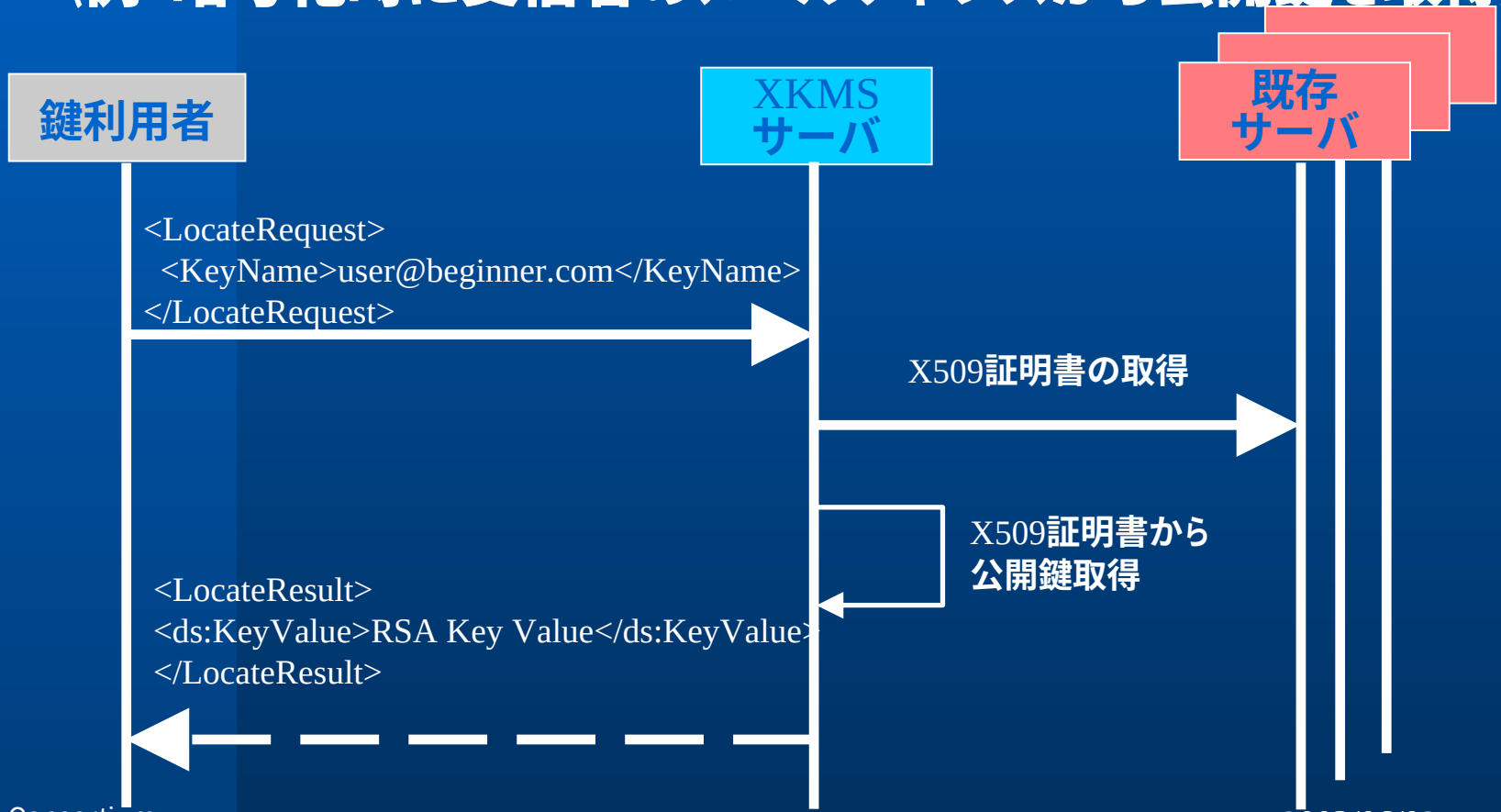
- 公開鍵関連情報の登録･失効･回復
  - クライアントサイド生成鍵の登録
  - サーバサイド生成鍵の取得



# X-KISS:Locate

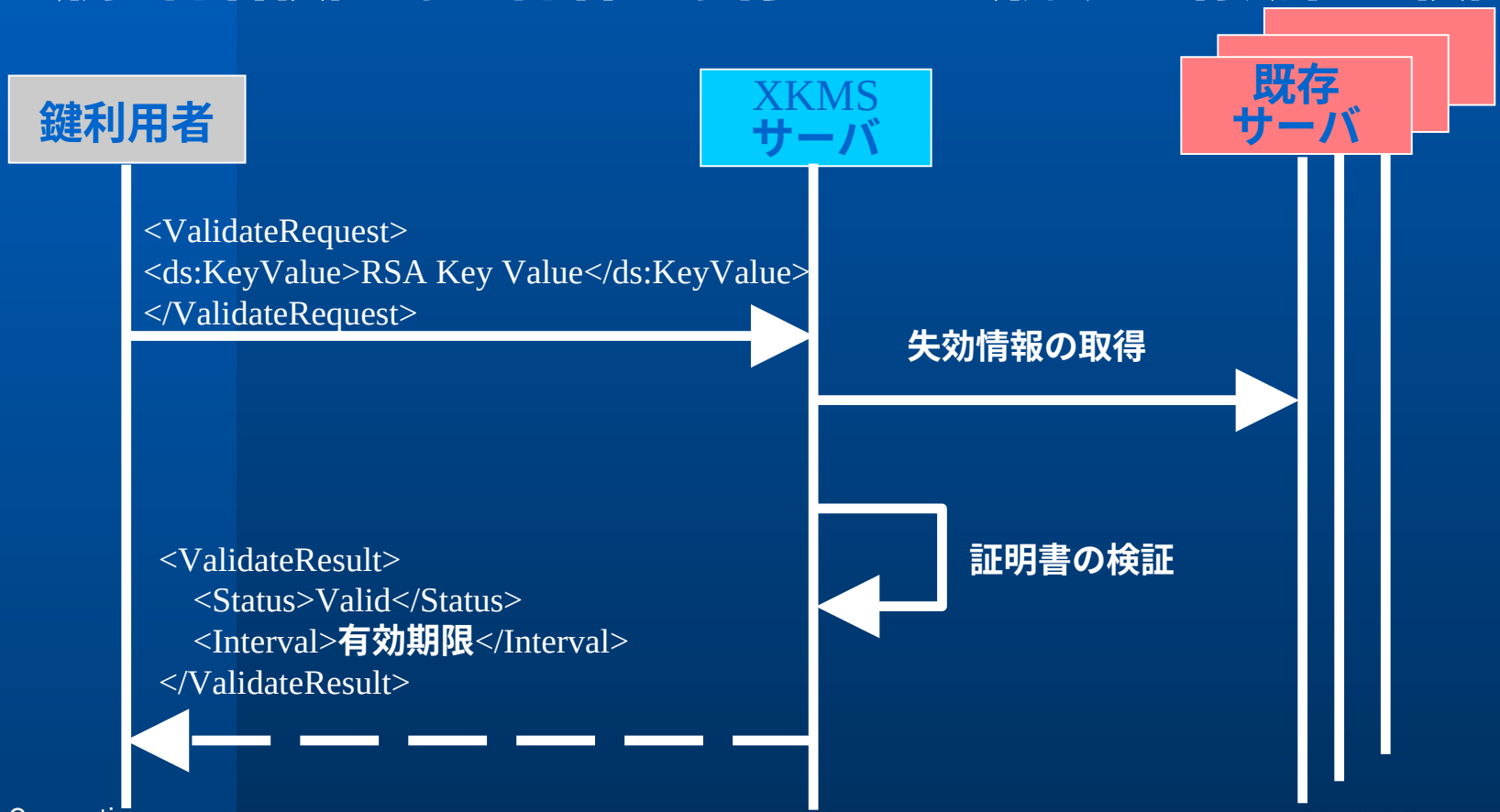
- **公開鍵関連情報を取得する**

**(例:暗号化時に受信者のメールアドレスから公開鍵を取得)**



# X-KISS:Validate

- **公開鍵関連情報の有効性を問合せる**
  **(例:署名検証時に署名に添付された公開鍵の有効性を検証)**



# 構文例（Validate Request）

```
<Validate Request xmlns="http://www.xkms.org/schema/xkms-2001-01-20">
  <Query>
    <Status>Indeterminate</Status>
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:KeyName>user@beginer.com</ds:KeyName>
      <ds:KeyValue>
        <ds:RSAKeyValue>
          <ds:Modulus>y0eZi+pL544O0anaCbHOF==</ds:Modulus>
          <ds:Exponent>AQAB</ds:Exponent>
        </ds:RSAKeyValue>
      </ds:KeyValue>
    </ds:KeyInfo>
  </Query>

  <Respond>
    <string>KeyValue</string>
    <string>ValidityInteval</string>
  </Respond>
</Validate Request>
```

Query：
公開鍵関連情報

Respond：
問合せ項目

# 構文例（ValidateResult）

```
<ValidateResult xmlns="http://www.xkms.org/schema/xkms-2001-01-20">
  <Reslut>Success</Reslut>
  <Answer>
    <KeyBinding>
      <Status>Valid</Status>
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:KeyValue>
          <ds:RSAKeyValue>
            <ds:Modulus>y0eZi+pL544O0anaCbHOF==</ds:Modulus>
            <ds:Exponent>AQAB</ds:Exponent>
          </ds:RSAKeyValue>
        </ds:KeyValue>
      </ds:KeyInfo>
      <ValidityInterval>
        <NotBefore>2000-09-20T12:00:00</NotBefore>
        <NotAfter>2002-09-20T12:00:00</NotAfter>
      </ValidityInterval>
    </KeyBinding>
  </Answer>
</ValidateResult>
```

Answer：
検証結果

# その他

- Interoperability
  - VeriSign,Entrustがテストサイトを公開
    （http://xmltrustcenter.org/index.htm）
- Availability
  - Toolkit
    - VeriSign(Java)
    - Entrust(Java)
    - Poupou(.NET)
    - Microsoft(ASP.NET)

# WS-Security

- **Web Service Security**

  **http://www.oasis-open.org/committees/wss/**

- **セキュリティ上の目標**
  - **セキュアなWebサービスの実現**
    - **トークンの受け渡し**
    - **完全性**
    - **秘匿性**

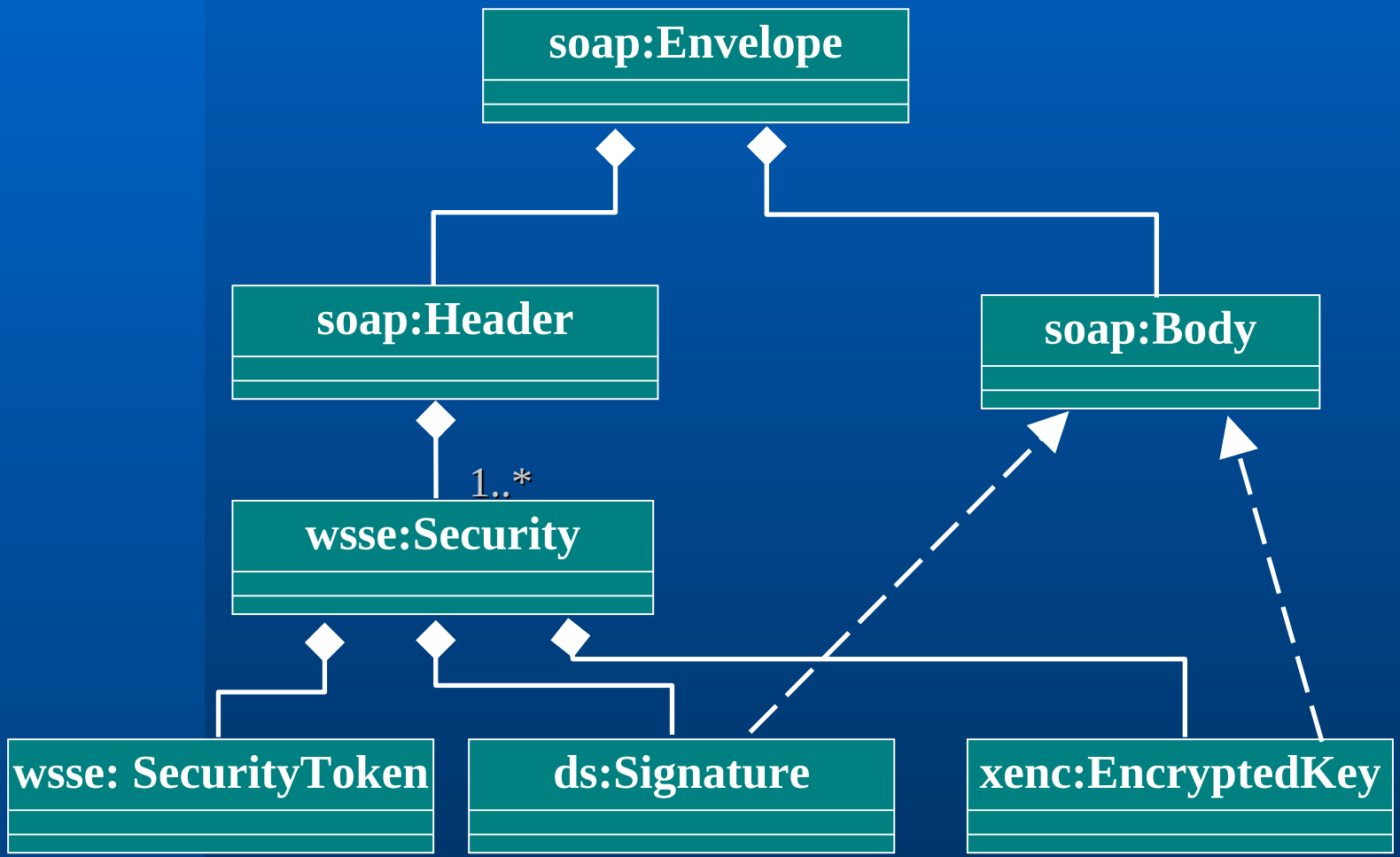
# WS-Security

- 経過
  - OASIS:Web Services Security TC
    - 2002年4月より活動開始
    - 2002年4月5日版 作業草案（7月にOASIS提出）
      （http://www-6.ibm.com/jp/developerworks/webservices/020607/j_ws-secure.html）
      2003年03月03日　Working Draft 11
    - メンバ:IBM,Microsoft,VeriSign…..(47社)
      - SOAP-SEC:Signature ー IBM,MS
      - SOAP-SEC:Encryption,Security Token ー IBM
      - WS-Security,WS-License ー IBM

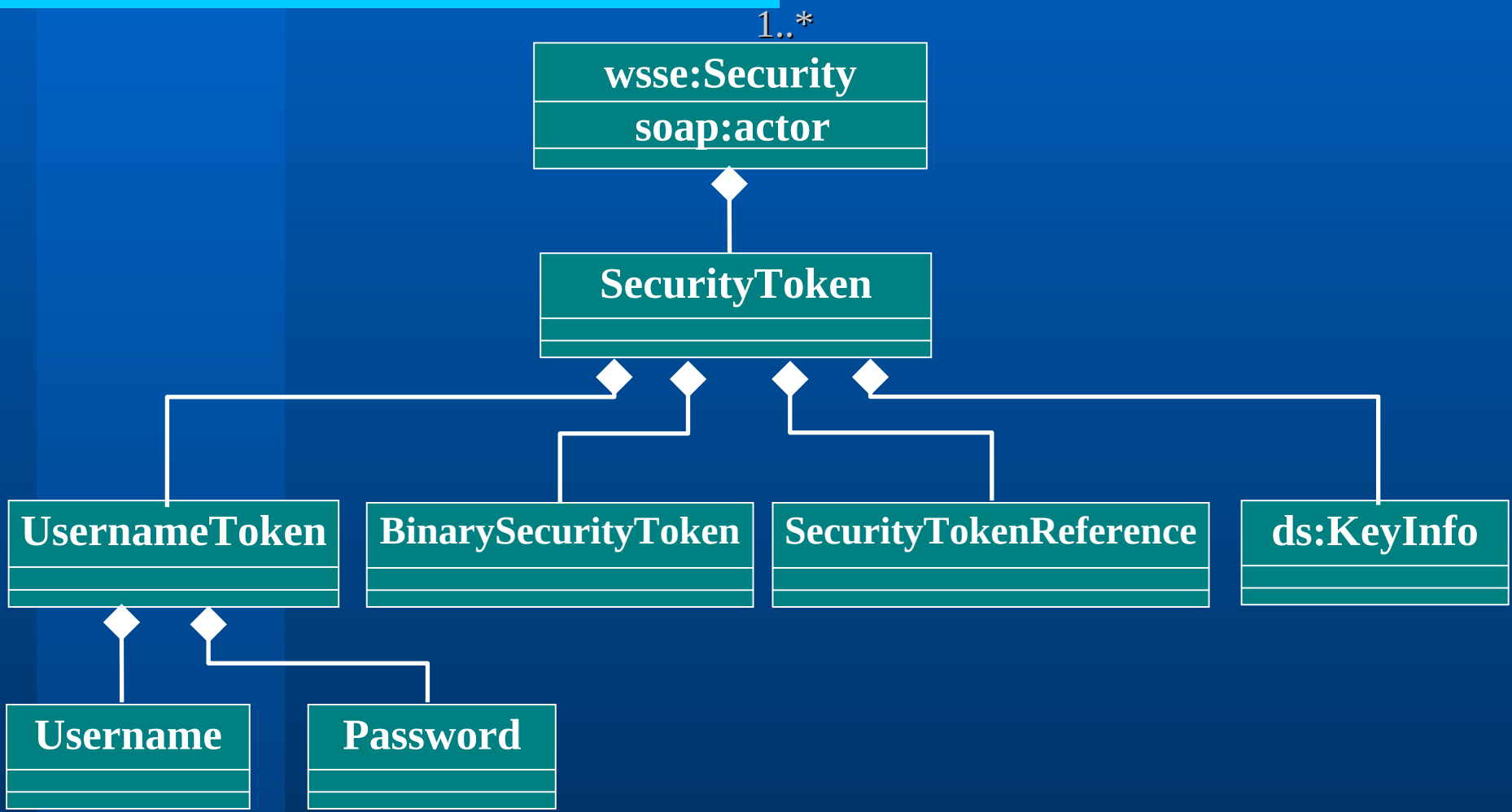
# WS-Security

- 特徴
  - Webサービスの完全性･秘匿性を実現するためのSOAP拡張メッセージ
    - セキュリティトークンの受け渡し
    - メッセージの完全性（XML Signature）
    - メッセージの秘匿性（XML Encryption）

# メッセージ構成

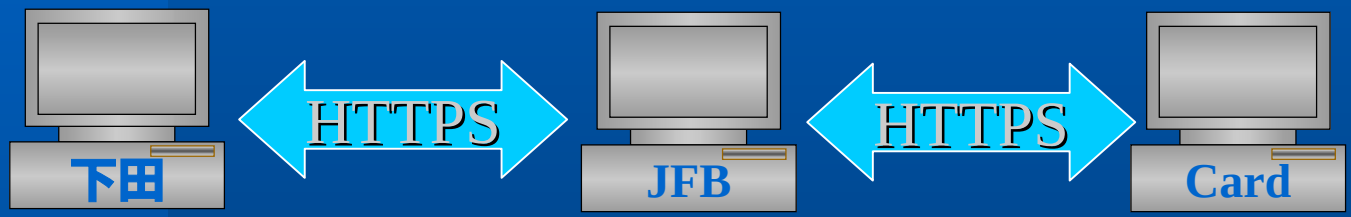


# メッセージ構成 (続)



# SSLとの比較

- SSL:Point to Point
  - 中継者へのセキュリティ確保が困難




- WS-Security:End to End

# Signature、Encryptionとの比較

- WS-Security:封筒＋割印
  - アプリケーションは意識しない


- Signature/Encryption:書類に捺印
  - アプリケーションが意識する

# その他

- Availability
  - Toolkit:
    - http://www.alphaworks.ibm.com/tech/webservicestoolkit
    - http://msdn.microsoft.com/webservices/building/wsdk/
- ロードマップ
  - WS-Security上に複数のモデルを提案予定
    - Policy、Trust、Privacy、SecureConversation、Federation、Authorization
    - IBM、Microsoft、VeriSign共同提案

      http://www-6.ibm.com/jp/developerworks/webservices/020607/j_ws-secmap.html

# SAML

- **Security Assertion Markup Language**

  **http://www.oasis-open.org/committees/security/**

- **セキュリティ上の目標**
  - **一度のログオン操作で複数のサイト/資源へのアクセス制御を可能にする（Single Sign-On)**
  - **サービスの意思決定に必要な属性情報の交換を可能にする**

    **→ 認証機能と認可機能を分散して配置可能**

# SAML（続）

- 経過
  - OASIS: XML-Based Security Services TC
    - 2000年12月より活動を開始
    - メンバー：Sun, HP, IBM, Entegrity, Oblix, …

      AuthML – Outlook, Securant, …

      S2ML –Netegrity, VeriSign, Commerce One, webMethod, …

      X-TASS – VeriSign
    - 2002年4月16日に委員会標準(1.0)の承認への投票
    - 2002年5月31日に委員会標準(1.0 revision 01)を提出
    - 2002年11月5日にOASIS標準
    - 2003年5月3日にSAML1.1が委員会標準作業草案

# SAML（続）

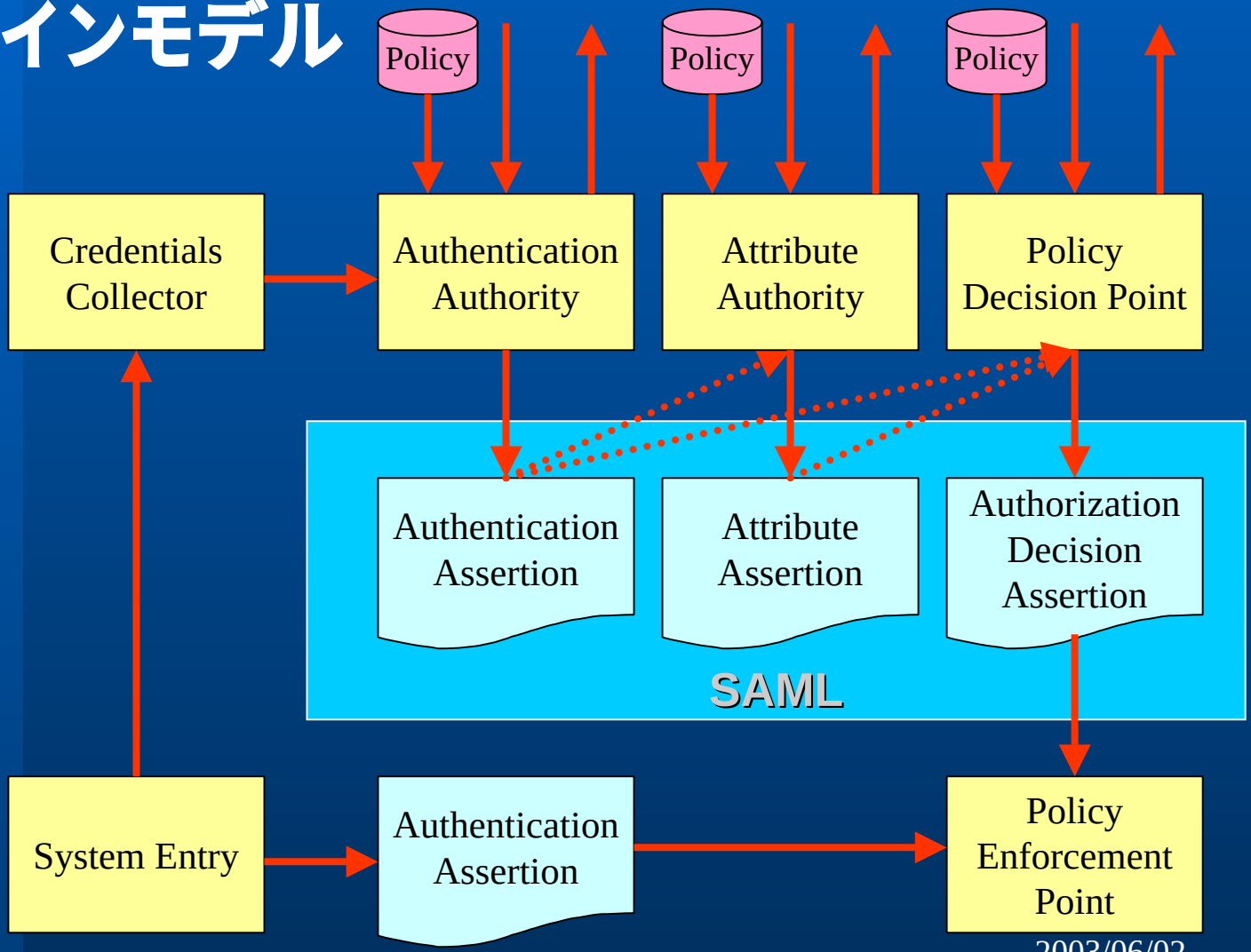
- 特徴
  - 協調モデルを用いたセキュリティ情報交換のためのフレームワーク
  - 既存認証機構との連携
    - Password, Kerberos, Secure Remote Password, Hardware Token, SSL/TLS Cert., X.509, PGP, …
  - 三つのモデル
    - Pull / Push / 3rd Party Security Model
  - SOAP/HTTPの特徴を生かしたプロトコル
    - Redirection, Proxy, …

# SAML（続）

- **SAML 1.0 Specification Set**
  - **Complete SAML v1.0 specification set**
    - **Assertions and Protocol**
      - **Assertion Schema**
      - **Protocol Schema**
    - **Binding and Profiles**
    - **Security and Privacy Considerations**
    - **Conformance Program Specification**
    - **Glossary**
  - **Draft**
    - **Profile document**
      - **WS-Security SAML Token Profile Draft6**

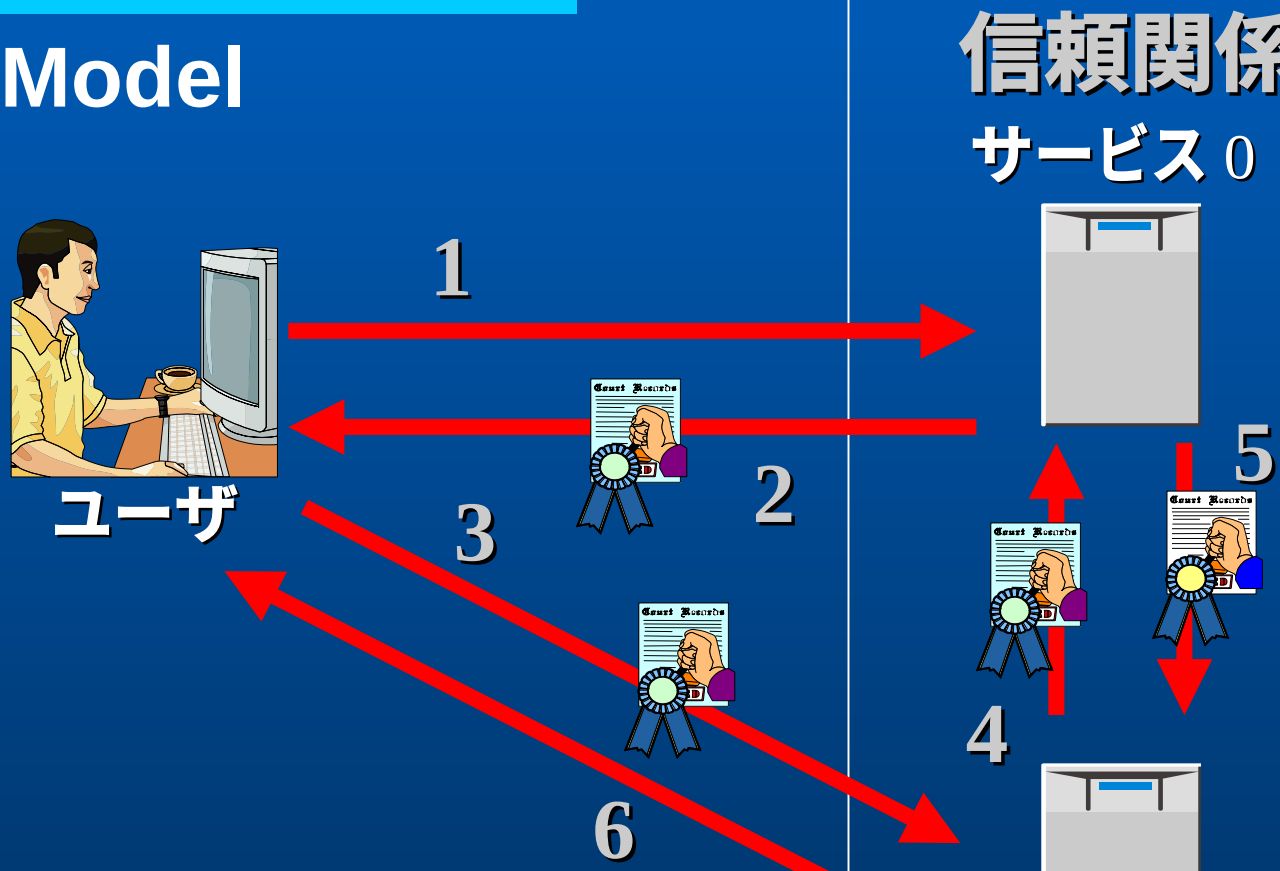
# SAML（続）

- ドメインモデル



# SAML（続）

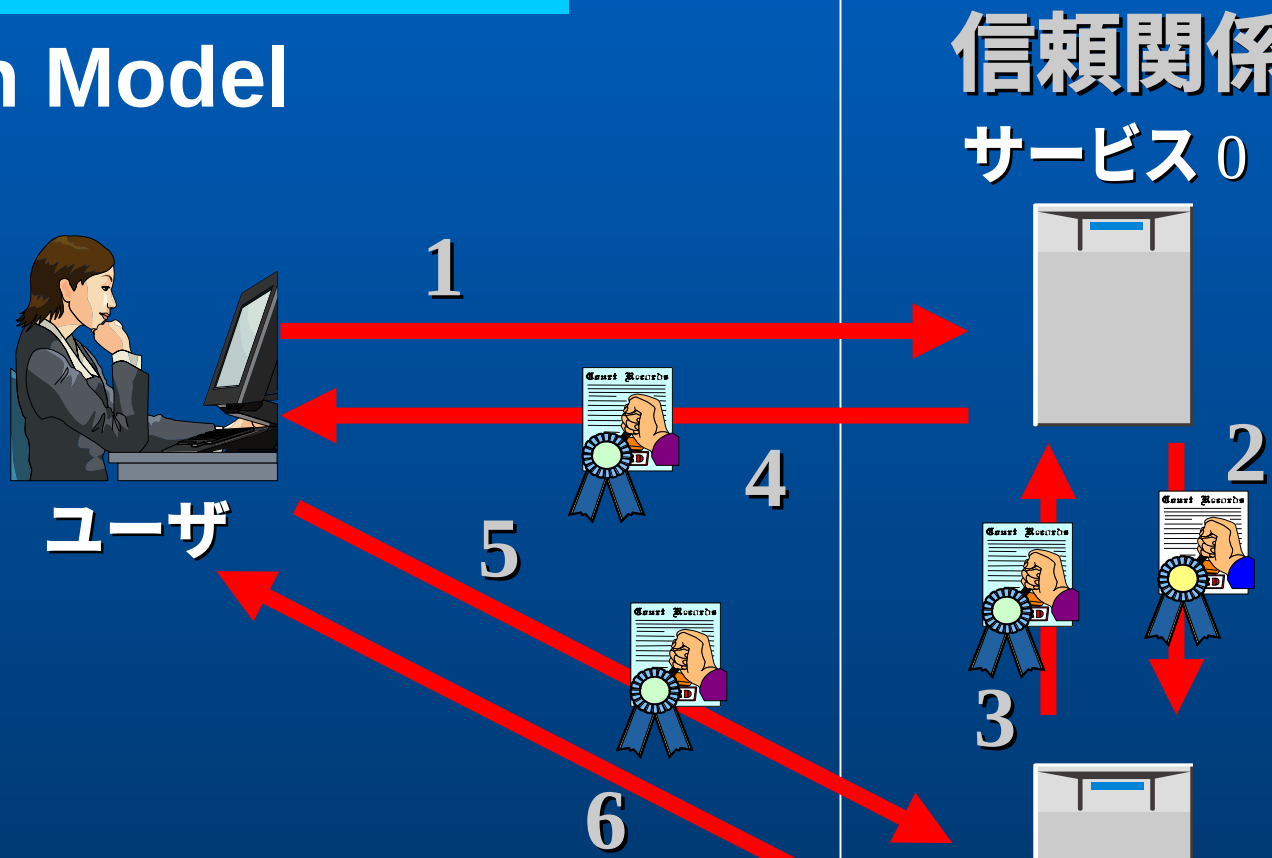
- Pull Model




サービスが認証情報をPull

# SAML（続）

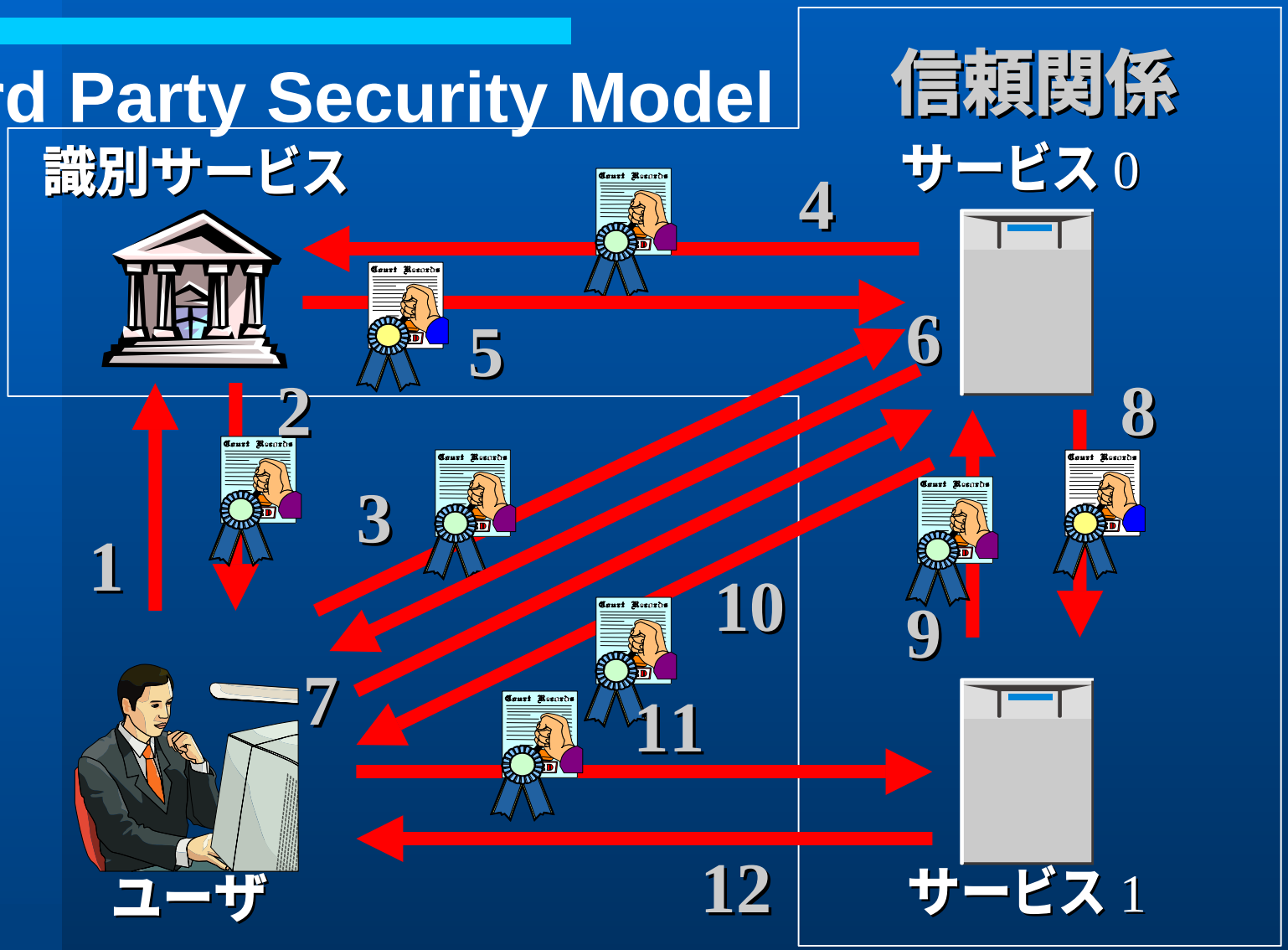
- Push Model




サービスが認証情報をPush

# SAML（続）

- 3rd Party Security Model



# SAML（続）

- 認証の要求メッセージ例

```
<samlp:Request MajorVersion="1" MinorVersion="0“
      RequestID=“8xtyzzKqPMLcFswefRIJAL">
      <samlp:RespondWith>AuthenticationStatement</samlp:RespondWith>
      <samlp:AuthenticationQuery>
        <saml:Subject>
          <saml:NameIdentifier Name=“JFB"/>
          <saml:SubjectConfirmation>
            <saml:ConfirmationMethod>
              http://www.oasis-open.org/…./draft-sstc-core-25/password
            </saml:ConfirmationMethod>
            <saml:SubjectConfirmationData>
              uTKaRyQmytsz=
            </saml:SubjectConfirmationData>
          </saml:SubjectConfirmation>
        </saml:Subject>
      </samlp:AuthenticationQuery>
</samlp:Request>
```

**NameIdentifier**
**名前による識別**

**ConfirmationMethod**
**パスワードによる確認**

# SAML（続）

- 認証の応答メッセージ(認証アサーション)

```
<samlp:Response InResponseTo="8xtyzzKqPMLcFswefRIJAL"
    MajorVersion="1" MinorVersion="0"
    ResponseID="xmlconsortium2002081002090011">
    <samlp:Status>
      <samlp:StatusCode Value="samlp:Success"/>
    </samlp:Status>
    <saml:Assertion AssertionID="qJcZsDTnJBPPe/4tlJKuZ/OLMtE= "
      IssueInstant="2002-06-10T11:22:33.456" Issuer="JUSTAir"
      MajorVersion="1" MinorVersion="0">
      <saml:Conditions
        NotBefore=" 2002-08-10T11:22:33.466"
        NotOnOrAfter=" 2002-08-10T15:22:33.466 "/>
      <saml:AuthenticationStatement
        AuthenticationInstant=" 2002-08-10T11:22:33.106"
        AuthenticationMethod="http://www.oasis-open.org/…/password">
        <saml:Subject>
          <saml:NameIdentifier Name="JFB" SecurityDomain="just:Reservation"/>
          <saml:SubjectConfirmation>
            <saml:ConfirmationMethod>
              http://www.oasis-open.org/…/password
            </saml:ConfirmationMethod>
          </saml:SubjectConfirmation>
        </saml:Subject>
      </saml:AuthenticationStatement>
    </saml:Assertion>
</samlp:Response>
```

Assertion
認証の証明

NameIdentifier
ユーザ識別

# SAML（続）

- Interoperability
  - OASISがInterOpでデモンストレーション
    - 12社が参加 (Portal Site / Contents Site)
- Availability
  - Liberty Alliance 1.1
  - 既に製品が出荷されている
    - 多くの会社が製品化/対応を表明
  - フリーな実装もある (Java / C++)
    - http://www.opensaml.org/ by Internet2(UCAID)
      - Shibbolethで採用
      - Modified Apache/BSD-style license

# SAML（続）

- その他
  - Java API標準の策定
    - JSR-155 Standard API for SAML
  - RSA Securityが基本特許(4件)を所有
    - RSAから無料で使用権がライセンスされる[2003/1]
  - 集中モデルから協調モデルへ
    - 信頼関係の構築が鍵

# XACML

- eXtensible Access Control Markup Language

  http://www.oasis-open.org/committees/xacml/

- セキュリティ上の目標
  - 任意の資源への詳細なアクセス制御のポリシーを表現する
    - 拡張性、相互接続性

    → 認証機能から独立した柔軟なアクセス制御

# XACML（続）

- 経過
  - OASIS: eXtensible Access Control Markup Language TC
    - 2001年4月16日より活動を開始
    - メンバー：Entrust, Entegrity, Crosslogix, IBM, …
      XACL – IBM
    - 2003年02月06日にOASIS標準として承認

# XACML（続）

- XACL – XML Access Control Language
  - IBMのXSS4Jに含まれる
  - XACMLのもととなった → XACLは柔軟性に課題

```
<xacl>
      <object href="//*[@member='premium']" />
      <acl>
        <subject><role>Requester</role></subject>
        <action name="read" permission="grant" />
        <condition operation="and">
          <predicate name="compareStr">
            <parameter value="eq" />
            <parameter>
              <function name="getUid" />
            </patameter>
            <parameter>
              <function name="getvalue">
                <parameter value="/name" />
              </function>
            </parameter>
          </predicate>
      </acl>
</xacl>
```

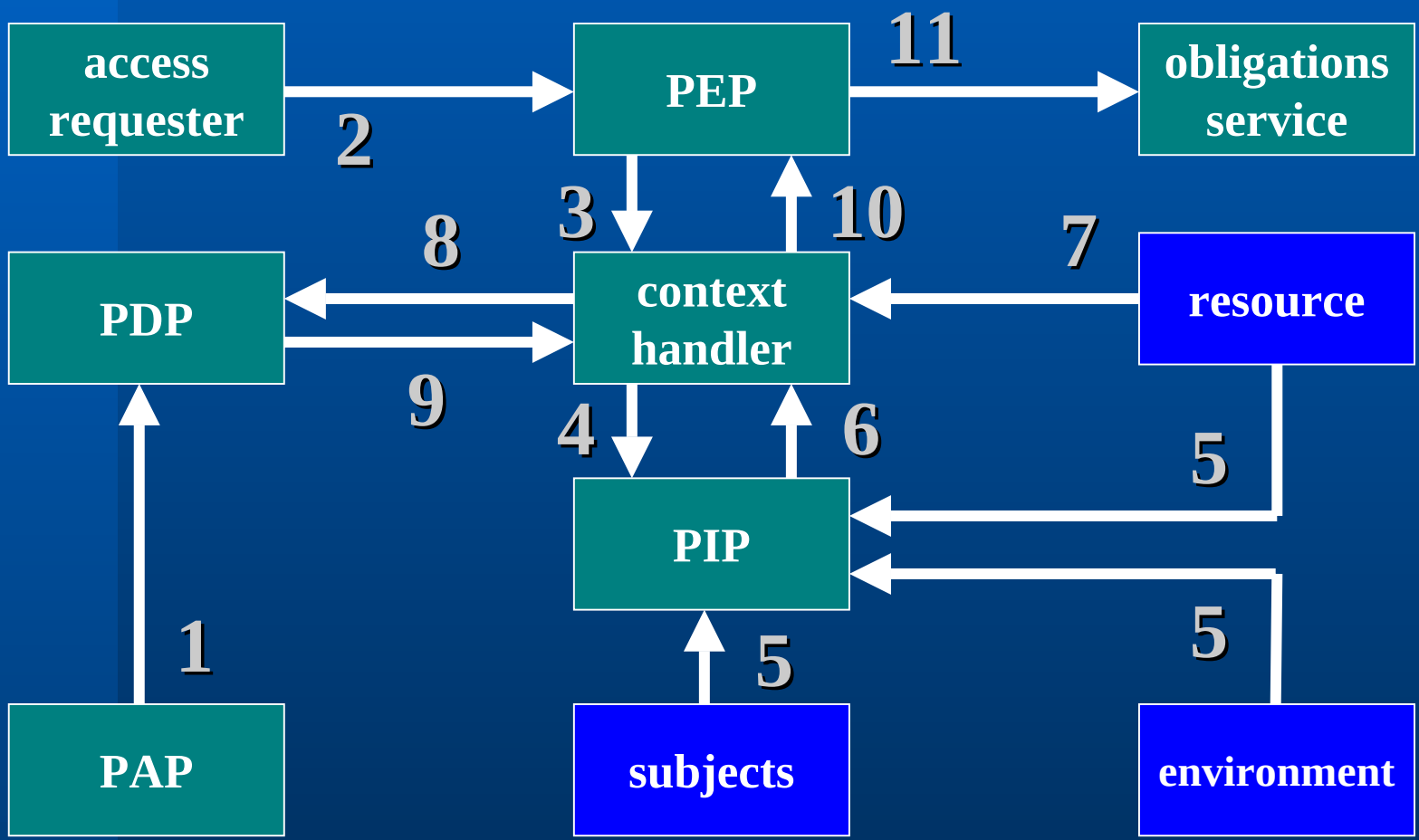
# XACML（続）

- 特徴
  - 認証機構から独立したアクセス制御ポリシーを表現するためのフレームワーク
  - 様々なリソースへのアクセス制御ポリシー表現
  - 簡単な関数や演算機能
- XACML Specifications
  - OASIS Standard [18 Feb. 2003]
    - Specification Document
    - Policy Schema
    - Context Schema

# XACML（続）

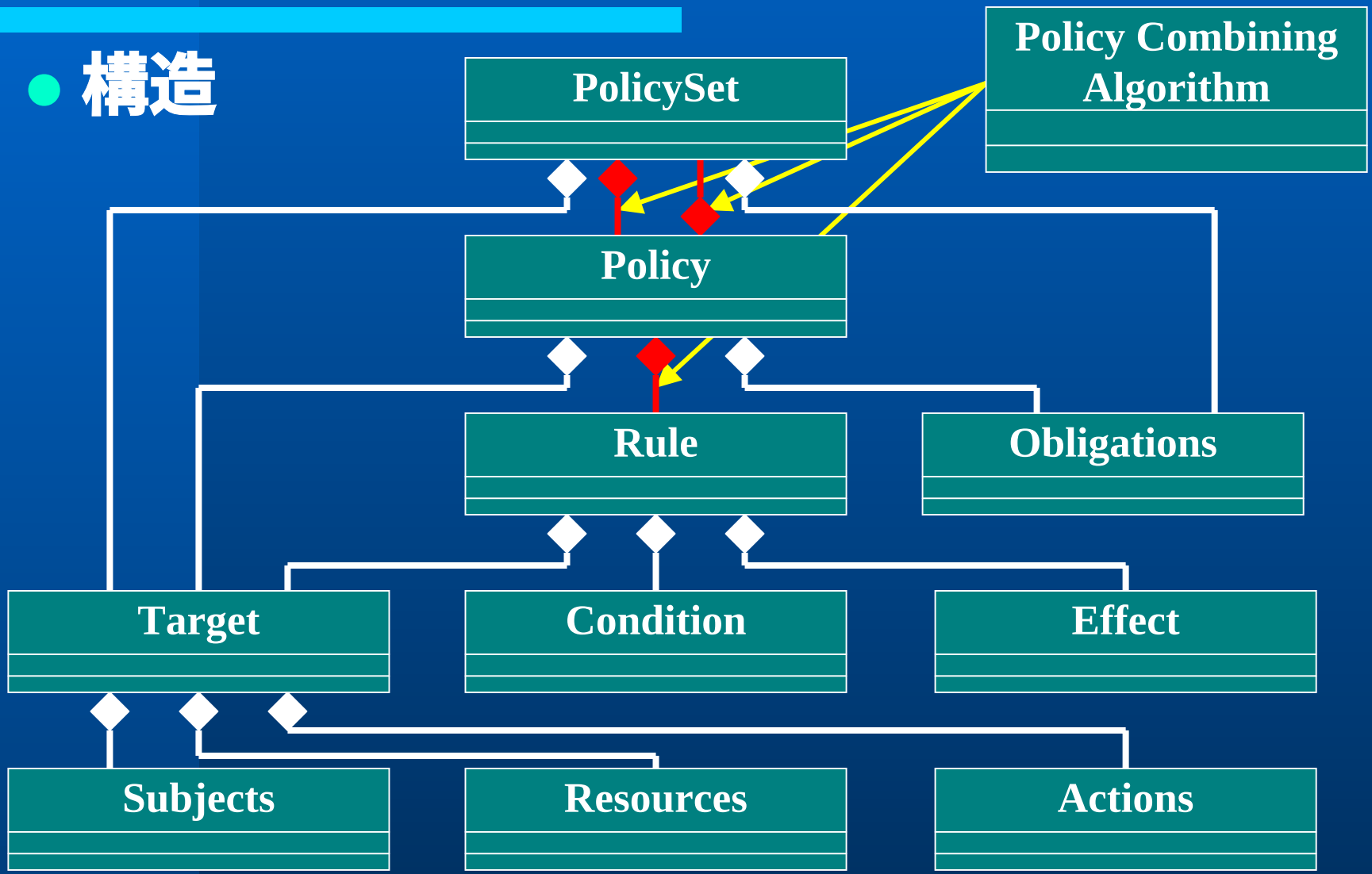
- データフローモデル

PEP: Policy enforcement point
PDP: Policy decision point
PAP: Policy administration point
PIP: Policy information point



# XACML（続）

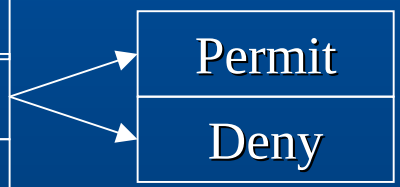
- 構造



# XACML（続）

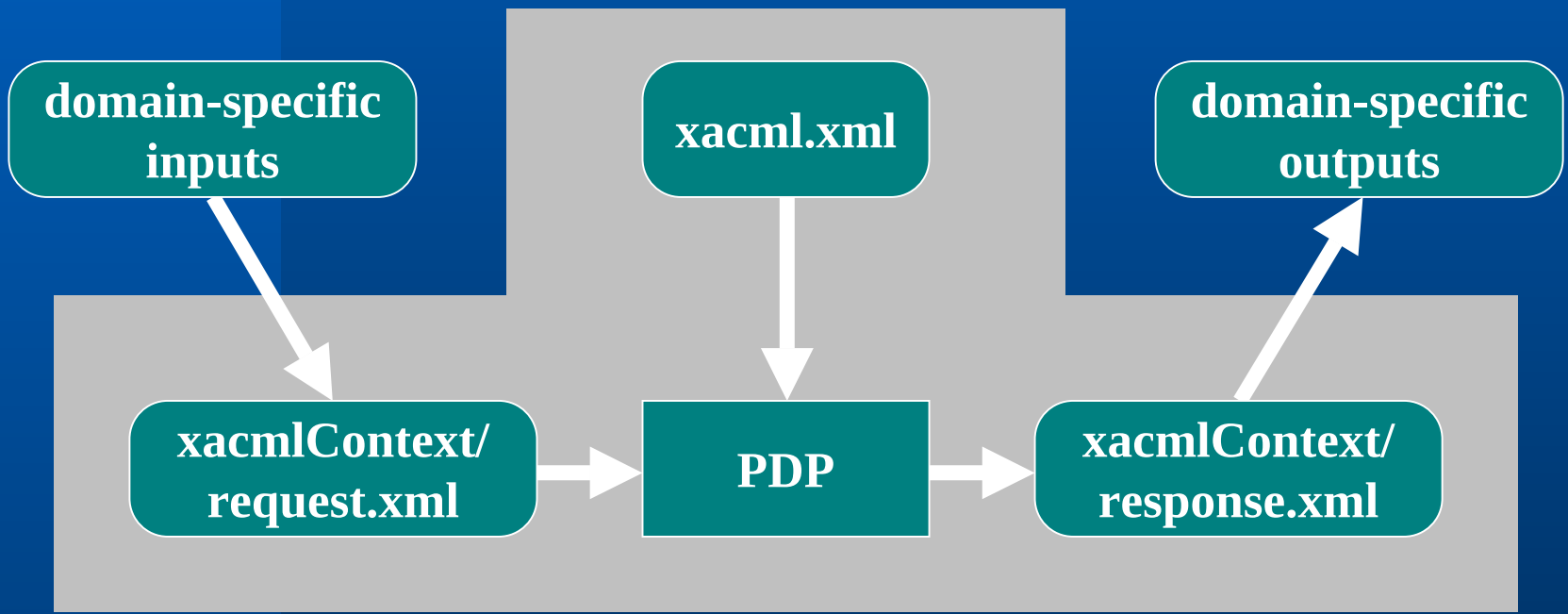
- ルールの評価
  - ルールを評価した結果、ルールが有効であった場合に、Effect属性に応じて、アクセスが「許可(Permit)」または「拒否(Deny)」となる

| Target | Condition | Rule |
|---|---|---|
| Match | True | Effect |
| Match | False | Not applicable |
| Match | Indeterminate | Indeterminate |
| No-match | True | Not applicable |
| No-match | False | Not applicable |
| No-match | Indeterminate | Not applicable |



# XACML（続）

- **XACML Context**
  - ポリシーを評価する際に参照するデータを抽象化



# XACML（続）

```
<Rule RuleId="//cons.com/rule/id/1" Effect="Permit">
    <Description>Sample policy</Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch MatchId="function:rfc822name-match">
            <SubjectAttributeDesignator AttributeId="identifier:subject:subject-id"
                                        DataType="identifier:datatype:rfc822Name" />
            <AttributeValue DataType="identifier:datatype:rfc822Name">*@xmlconsortium.org</AttributeValue>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>  <AnyResource />  </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="function:string-equal">
            <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action" DataType="xs:string" />
            <AttributeValue DataType="xs:string">read</AttributeValue>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition FunctionId="function:dayTimeDuration-greater-than">
      <Apply FunctionId="function:date-substract">
        <EnvironmentAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:env:date" DataType="xs:date" />
        <AttributeSelector RequestContextPath="/ctx:Request//ctx:ResourceContent/ed:employee/ed:DoB"
                           DataType="xs:date" />
      </Apply>
      <AttributeValue DataType="xs:dayTimeDuration">20-0-0</AttributeValue>
    </Condition>
</Rule>
```

# XACML（続）

- **Request Context**

```
<?xml version="1.0" encoding="UTF-8"?>
<Request
      xmlns="urn:oasis:names:tc:xacml:1.0:context"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:context
        http //www.oasis-open.org/tc/xacml/1.0/sc-xacml-schema-context-01.xsd">
      <Subject>
        <Attribute
            AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
            DataType="identifier:rfc822name">
          <AttributeValue>michimura@xmlconsortium.org</AttributeValue>
        </Attribute>
      </Subject>
      <Resource>
        <Attribute
            AttributeId="identifier:resource:resource-uri"
            DataType="xs:anyURI">
          <AttributeValue>http://cons.com/record.txt</AttributeValue>
        </Attribute>
      </Resource>
      <Action><
        <Attribute
            AttributeId="urn:oasis:names:tc:xacml:1.0:action"
            DataType="xs:string">
          <AttributeValue>read</AttributeValue>
        </Attribute>
      </Action>
</Request>
```

# XACML（続）

- **Response Context**

```
<?xml version="1.0" encoding="UTF-8"?>
<Response
      xmlns="urn:oasis:names:tc:xacml:1.0:context"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:oasis:names:tc:xacml:1.0:context
        http //www.oasis-open.org/tc/xacml/1.0/sc-xacml-schema-context-01.xsd">
      <Result>
        <Decision>
          Deny>
        </Decision>
      </Result>
</Request>
```

- **特許情報**
  - **IBMが2件を所有**
  - **ContentGuardが5件を所有**
    - **使用権に関する態度は表明していない[2002年9月現在]**

# Links

- XML Signature:
  - http://www.w3.org/TR/xmldsig-core/
- XPath Filter 2.0:
  - http://www.w3.org/TR/xmldsig-filter2/
- Exclusive XML Canonicalization
  - http://www.w3.org/TR/xml-exc-c14n/
- XML Encryption:
  - http://www.w3.org/TR/xmlenc-core/
- Decryption Transform:
  - http://www.w3.org/TR/xmlenc-decrypt/

# Links (続き)

- WS-Security
  - http://www-106.ibm.com/developerworks/library/ws-secure/#references
  - http://www-6.ibm.com/jp/developerworks/webservices/020607/j_ws-secure.html
- XKMS 2.0
  - http://www.w3.org/TR/xkms/
- SAML
  - http://www.oasis-open.org/committees/security/
- XACML
  - http://www.xacml.org