Webサービスと
Webサービスのセキュリティ

2003.12.5　Internet Week

鈴木優一
エントラストジャパンCTO

目次

- Webサービス
  - Webサービスの基本
    - Webサービス、SOAP、WSDL、UDDI
  - Webサービスのデモ 
- Webサービスのセキュリティ
  - Webサービスのセキュリティ基本
    - XML署名、XML暗号、XKMS
  - XMLセキュリティの基本デモ（XML署名と署名検証）
  - Webサービスのセキュリティ応用
    - SAML、XACML、Liberty、WSS、DSS

Webサービス

- Webサービス：分散システム
- SOAP（XMLプロトコル、RPC)
- WSDL：Webサービス・インタフェース定義
- UDDI：Webサービスの登録と検索

今までの分散システム

- DCE RPC（Remote Procedure Call)
  - OSF 1980年代後半
  - プロシージャ
  - DCE環境に閉じる、成功しなかった
- CORBA IIOP (Internet Inter-ORB Protocol)
  - OMG 1990年代半ば
  - 分散オブジェクト
- MS DCOM (Distributed Common Object Model)
  - Windowsの世界に閉じる
- いずれも共通の環境を前提としている
- TCP Socket上のヘビーなプロトコル

Webサービス：疎結合な分散システム

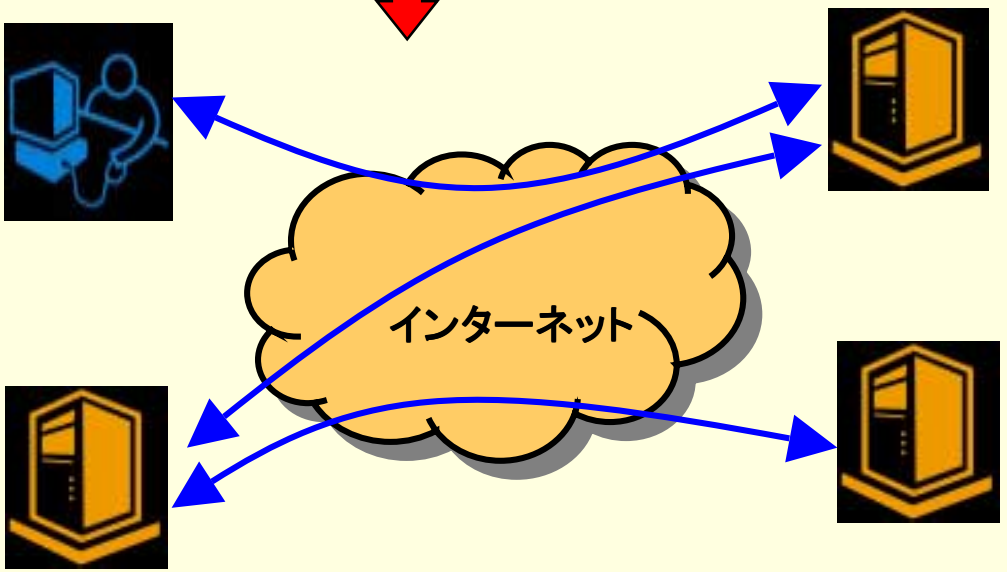
クライアント／サーバ

- 言語依存：C/C++
- 蜜結合
- イントラネット
- TCPソケット

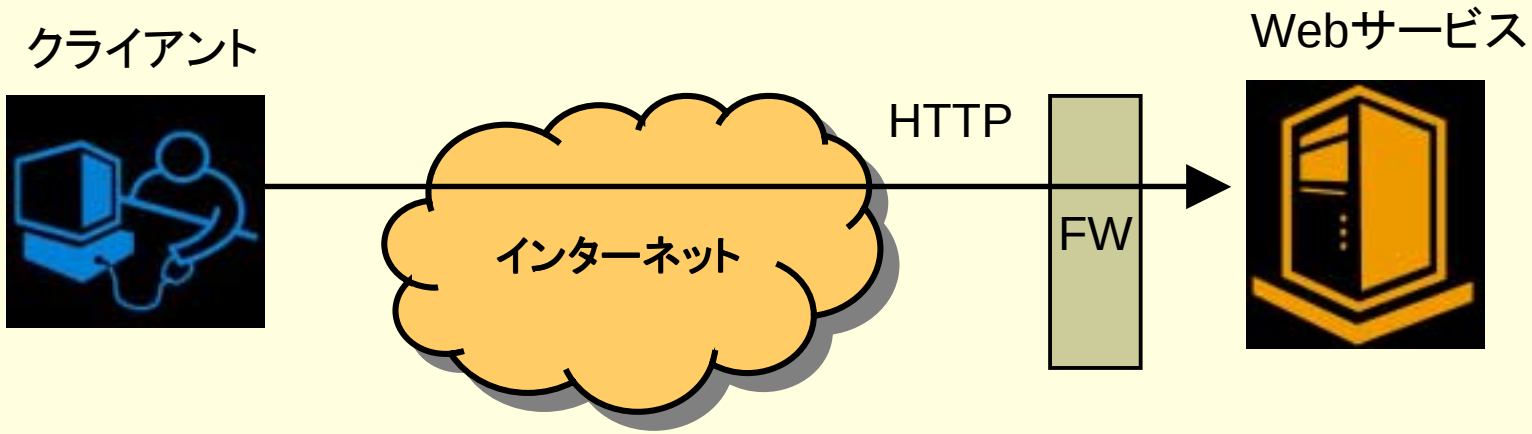
分散システムのパラダムシフト

Webサービス

- 疎結合
- プラットフォーム非依存
- 言語非依存
- インターネット
- XML / SOAP / HTTP



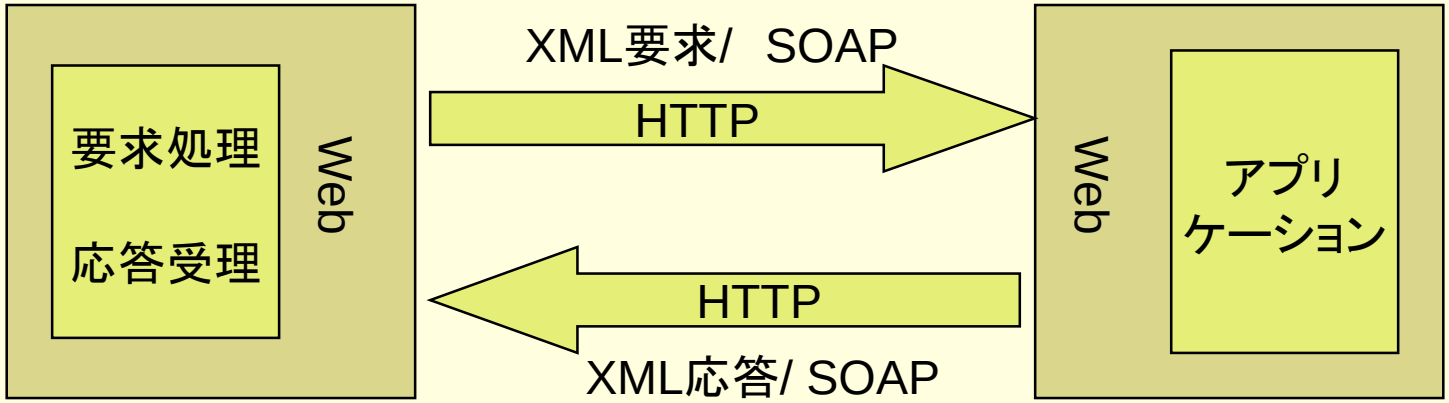
Webサービス：ファイアウォール・フレンドリ



HTTPでファイアウォール
を通過できる

Webサービス

- OS、アプリ開発言語に依存しない

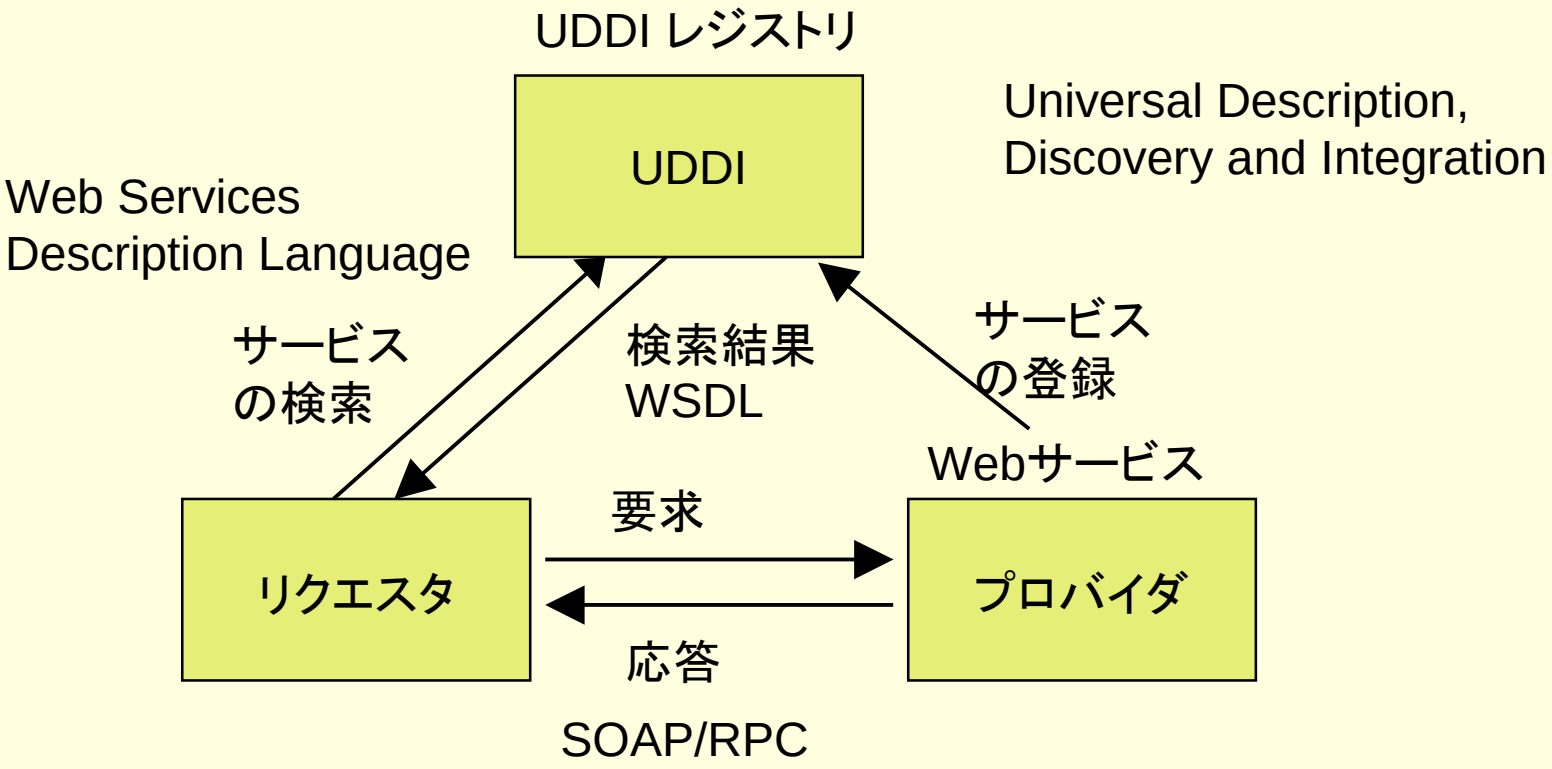


例：OS UNIX
言語 C

例：OS Windows
言語 VB

Webサービスのフレームワーク

- Webサービスの関係



Webサービス

- W3C　Webサービス活動
  - http://www.w3.org/2002/ws/
  - Webサービス・アーキテクチャ
  - WSDL1.2
  - Webサービス・プロトコル
    - SOAP1.2

SOAP ：XMLメッセージ交換プロトコル

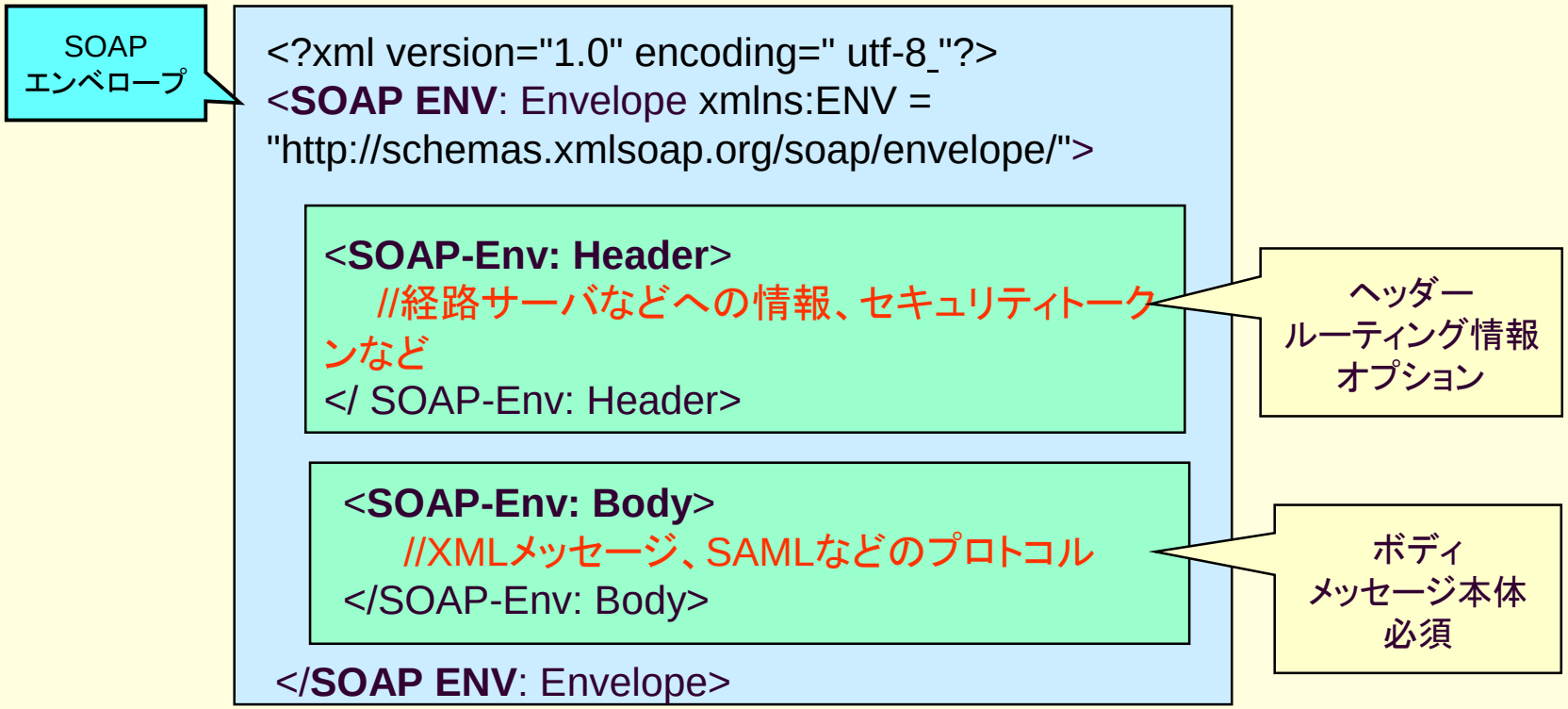
- SOAP（Simple Object Access Protocol）1.2
  - W3C Recommendation 24 June 2003
  - 標準化団体：W3C
    - http://www.w3.org/TR/2003/PR-soap12-part1-20030624/
  - XMLベースのメッセージ交換プロトコル（**RPC**)
- SOAP：Webサービスで使用されるメッセージのデータフォーマットや、メッセージの処理ルールを定めた通信規約、エンベロープの仕様
- SOAP 1.1(W3C Note)からRecommendation
  - V1.2 はv1.1の曖昧さを除去
  - 現在、多くの実装がv1.1で行われている
- SOAP 1.2 仕様
  - SOAP Version 1.2 Part 0: Primer
  - SOAP Version 1.2 Part 1: Messaging Framework
  - SOAP Version 1.2 Part 2: Adjuncts
  - SOAP Version 1.2 Specification Assertions and Test Collection

SOAPの特徴

- 分散オブジェクトにアクセスするためのメッセージ交換規約
  - メッセージのエンベロープ
  - ヘッダーとボディ
- IIOPなどが行うトランザクション管理はしない
  - 軽量、柔軟、祖結合
- トランスポートプロトコルとは独立
  - HTTP
  - SMTP
- SOAP処理に各種のToolが提供されている

SOAP エンベロープ

- SOAPエンベロープの構造



SOAP over HTTP（要求）

- Example 1 SOAP メッセージ要求をHTTPに載せる

```
POST /StockQuote HTTP/1.1
Host: www.stockquoteserver.com
Content-Type: text/xml+soap; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
```

HTTP

```
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
 SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePrice xmlns:m="Some-URI">
      <symbol>DIS</symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

価格を問合わせ

SOAP
メッセージ

SOAP over HTTP（応答）

- Example 2　SOAPメッセージ応答をHTTPに載せる

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
```

HTTP

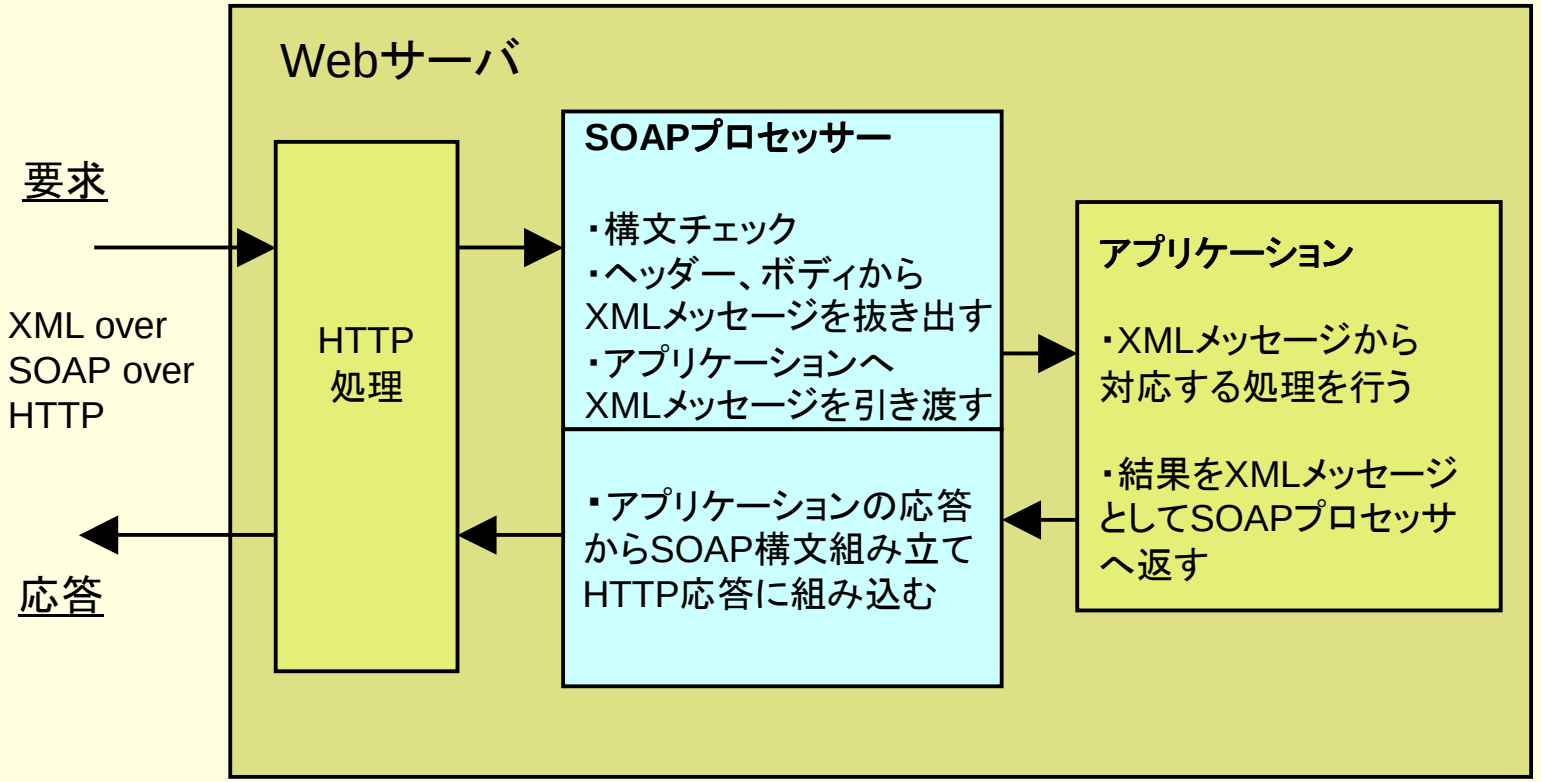
```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse xmlns:m="Some-URI">
      <Price>34.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

価格を答える

SOAP
メッセージ

SOAPプロセッサー

- SOAP RPC



SOAP RPC

- RPC リモートプロシージャコール
  - 例: int GetPrice (string name)
  - SOAP 要求（メソッド呼び出し）
    - メソッド名を要素名に
    - パラメータは子要素に

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope   xmlns:soapenv="http://www.w3.org/2001/12/soap-envelope">
 <soapenv:Header>
   ……
 </soapenv:Header>
 <soapenv:Body>
  <r:GetPrice ←メソッド名に対応する要素名
      soapenv:encodingStyle="http://www.w3.org/2001/12/soap-encoding"
      xmlns:r="http://example-market.co.jp/2003/10/quotes">
     <r:name>MIKAN</r:name> ←パラメータ名から取られた要素名
  </r:GetPrice>

 </soapenv:Body>
</soapenv:Envelope>
```

SOAP RPC 応答

- 応答結果は<rpc:result>要素で返される

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    ……
  </soapenv:Header>
  <soapenv:Body>
    <r:GetPriceResponse soapenv:encodingStyle=
                        "http://www.w3.org/2001/12/soap-encoding"
        xmlns:r=http://example-market.co.jp/2003/10/quotes
        xmlns:rpc=http://www.w3.org/2002/12/soap-rpc>
      <rpc:result>r:price</rpc:result>
      <r:price>100</r:price>    ←応答結果
    </r:GetPriceResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

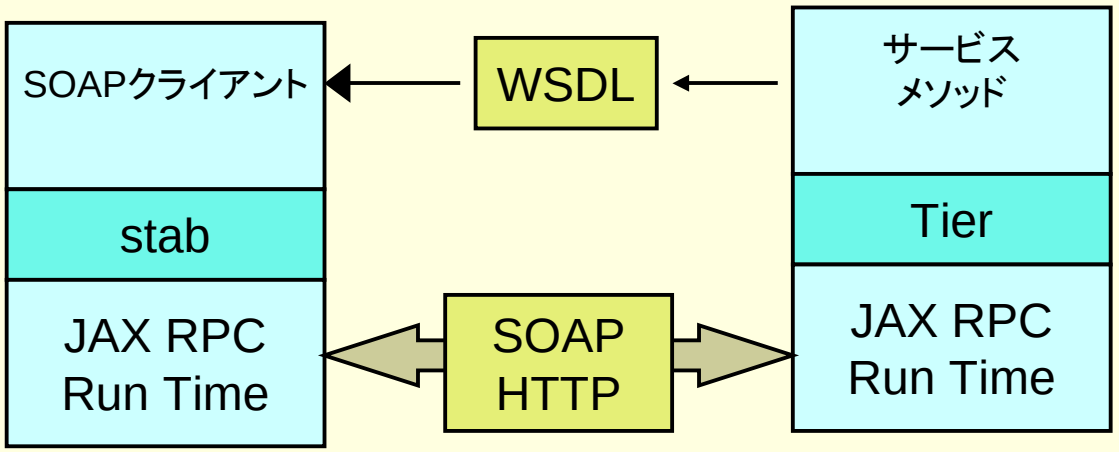
応答要素名

SOAP処理

- SOAP処理
  - SOAP構文解析
  - エラー処理：SOAPフォールト
    - <Fault>
      - Code　違反コードを記述する
      - Reason 人間が読めるエラー解説を記述する
      - Node　エラー発生元のURIを記述する
      - Detail　SOAP本体に関係するアプリケーション固有のエラー情報を記述する
  - XMLメッセージの抜出し、アプリケーションへ引渡し
  - 結果のSOAP構文組み立て、HTTPヘッダ埋込み

SOAP処理Tool

- 各種のToolが利用できる
  - Apache　AXIS 1.0
    - http://ws.apache.org/axis/index.html
    - SOAP 1.1/1.2対応
    - WSDLからJava　Code生成
    - SUN JAX-RPC互換
  - SUN JAXP, JAX-RPC



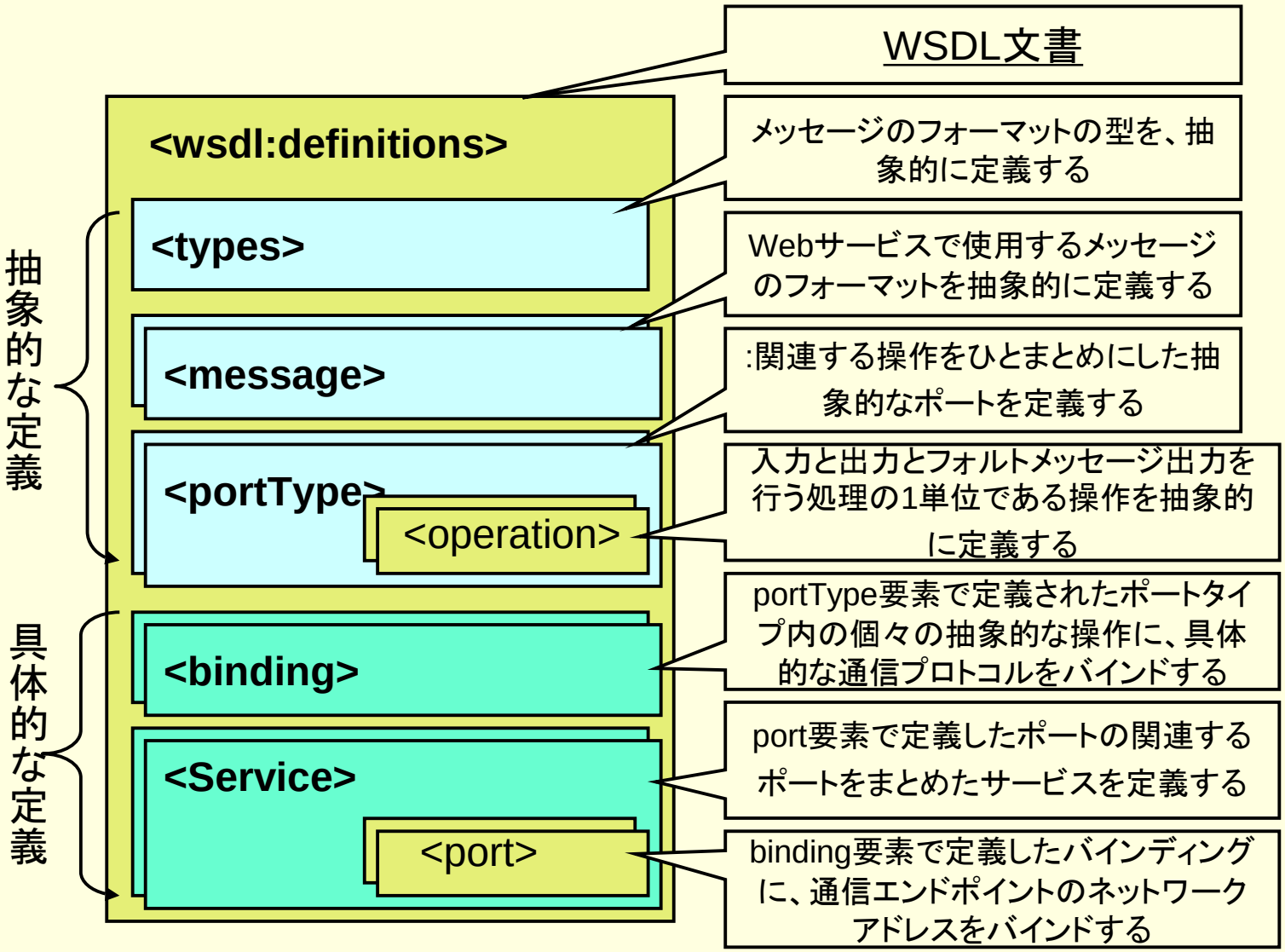
WSDL（Webサービス記述言語）

- WSDL (Web Service Description Language)
- WSDL 1.1
  - IBM, Microsoftらが W3C Noteとして公開
- WSDL 1.2　W3C Recommendationを目指して現在Draft策定中（2003末に標準へ）
  - CORBAのIDL（Interface Description Language)に相当するインタフェース記述仕様

WSDL (Web Service Description Language)

- Webサービスへのインタフェース記述言語
  - CORBAのIDL相当
- 何を記述するのか？
  - Webサービスの所在
  - Webサービスのメッセージフォーマット（入出力の定義）
  - Webサービスが用いる通信プロトコル
- WSDLプロセッサでメソッドのインタフェースを自動生成
  - →開発者はメッソド呼出しの中身のみを書けばよい
  - →自動生成の可能性もある
- WSDLの用途
  - Webサービスのインタフェースを公開することで利用者が簡単にWebサービスを利用できるようにする

WSDLの構造

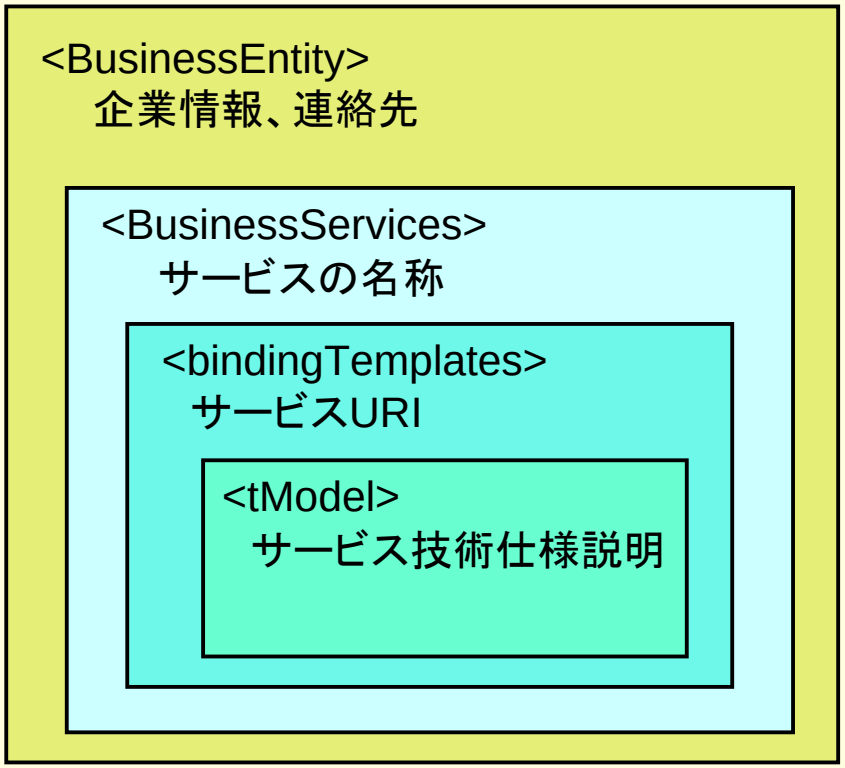


UDDI (Webサービスの登録と検索）

- UDDI (Universal Description, Discovery and Integration)
- UDDI.org v1.0 (2000)
  - http://uddi.org/
- OASIS UDDI v2.0、v3.0（19 July 2002 ）
  - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec
- Webサービス提供者の情報を登録公開
- 利用者が検索、サービスを受ける
  - パブリックUDDI
    - 誰でもが登録できる、セキュリティが必要
  - プライベートUDDI
    - クローズな企業内、企業間で登録

UDDIモデル

- Webサービス案内、説明



UDDI API

- UDDI検索、登録API
  - **発行API**
    情報の登録や削除、ユーザー認証を行うためのAPI
  - **照会API**
    情報の検索や詳細情報の取得を行うためのAPI
  - **セキュリティ方針API**
    認証トークンに関するAPI
  - **管理および所有権移動API**
    businessEntityまたはtModel構造の管理や所有権を移動するためのAPI
  - **サブスクリプションAPI**
    UDDIレジストリに加えられた変更を通知するためのAPI
  - **値セットAPI**
    登録時に呼び出されたkeyedReferenceの値が有効かどうかを検査するためのAPI

Webサービスの相互運用性

- WS-I
  - http://ws-i.org/
- Basic Profile 1.0 specification 2003/08/08
- 基本的相互運用性のためのプロファイル規定
  - **WS-Basic1**
    - XML Schema 1.0
    - SOAP 1.1
    - WSDL 1.1
    - UDDI 2.0
  - サンプル、テストToolの提供

Webサービスのデモ

- Apache AXISを用いたSOAPバインディング
  - サンプルクライアントプログラム
  - サンプルサーバプログラム
  - Webサービスの動作確認
- XML over SOAP over HTTP
  - リクエスト／レスポンスのトレース

- 伊藤　康宏
- エントラストジャパン

Webサービスのセキュリティ基本

- XML署名
- XML暗号
- XKMS（XML鍵管理サービス）

Webサービスのセキュリティ
標準化団体 W3CとOASIS

- W3C（http://www.w3.org/）
  - XMLベースのセキュリティ標準策定：**基本仕様**
  - XML ディジタル署名(**RFC 3075**)
  - XAdES；長期署名フォーマット
  - XML暗号
  - XKMS（XMLベースの鍵管理）
  - SOAP（Simple Object Access Protocol）
- OASIS（http://www.oasis-open.org/）
  - Organization for the Advancement of Structured Information Standards（XMLベースの業界標準化団体）：**応用仕様**
  - SAML（Security Assertion Markup Language）
  - XACML（XML Access Control Markup Language）
  - WSS (Web Service Security)
  - DSS (Digital Signature Services)

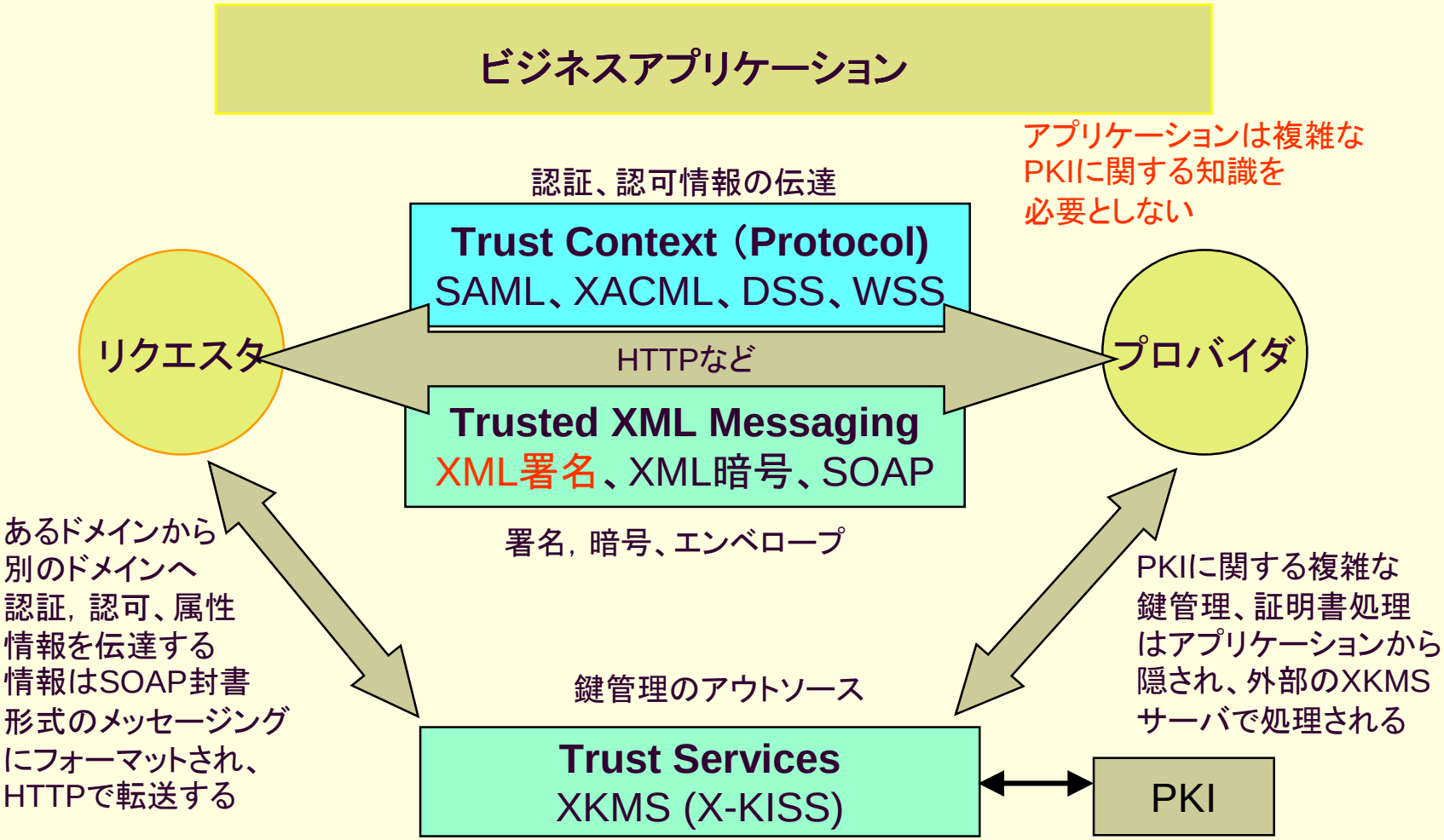
Webサービスのセキュリティ

ビジネス・アプリケーションの基盤へ

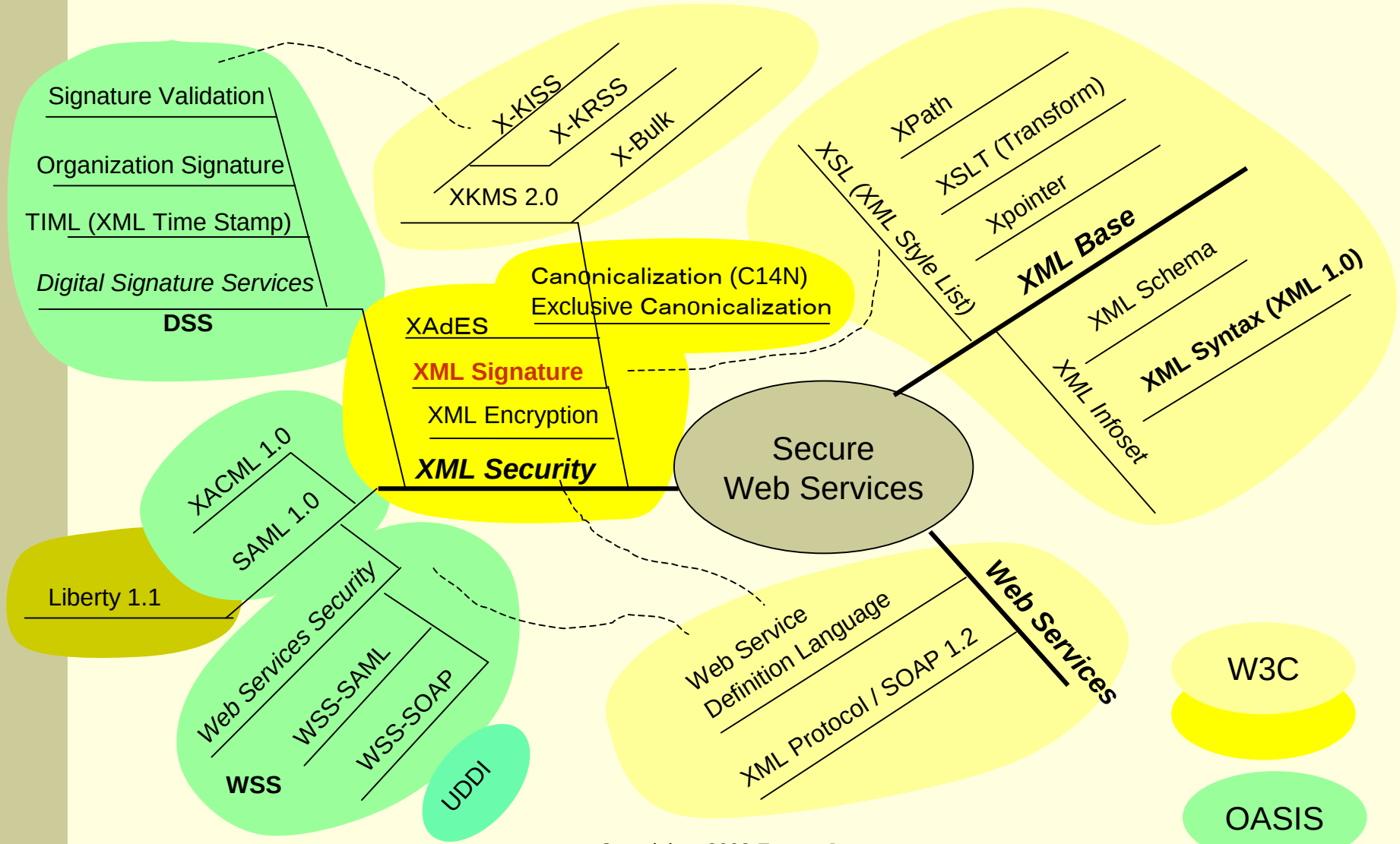
- Webサービス・セキュリティの必要性
  - インターネットでのWebサービス
  - セキュリティ・メカニズムが必須
  - 改ざん検出、発信源の特定
- セキュリティ・プロトコルの整備：PKIの利用
  - XMLディジタル署名
  - XML暗号
  - XMLメッセージング・プロトコル
  - Webサービスセキュリティ標準がほぼそろう

- 参考：Webサービスのセキュリティの解説（鈴木）
  - @IT Security Forum
  - http://www.atmarkit.co.jp/fsecurity/rensai/webserv03/webserv01a.html

Webサービスのセキュリティ・フレームワーク
–信頼できるXMLメッセージング–



Web サービス・XMLセキュリティ相関図



XMLディジタル署名ーXMLセキュリティの基盤

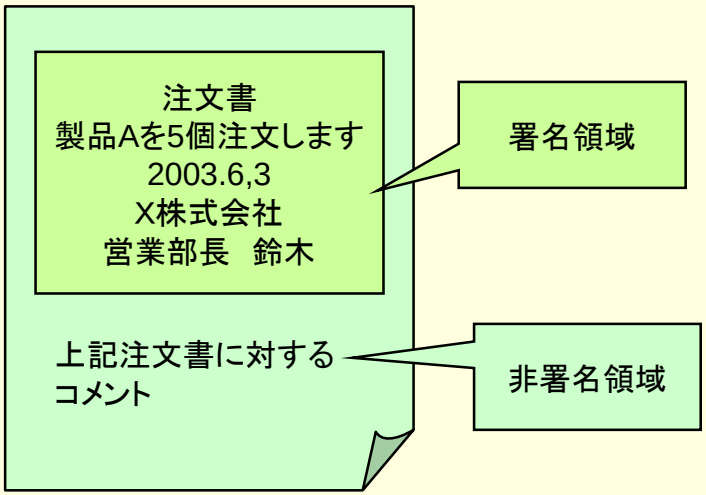
- 標準化: W3CとIETFの共同作業
  - XML-Signature Syntax and Processing
  - W3C http://www.w3.org/Signature/ Recommendation (2002.2)
  - IETF http://www.ietf.org/html.charters/xmldsig-charter.html
    - XML Digital Signature: RFC3075
- XML構文でディジタル署名タグを規定
  - CMS SignedData（ASN.1) に較べ署名の可読性を増す
  - XML文書との親和性
  - 電子申請フォームなどに適用
- 署名の前にXML文書の正規化(C14N)が必要
  - XML記述の任意性(空白、改行が任意に入れられる)を除く
    - Canonical XMLVersion 1.0 W3C Recommendation (2001.5)
      - RFC3076
    - Exclusive XML Canonicalization 1.0 W3C Recommendation (2002.6)

XML署名の例

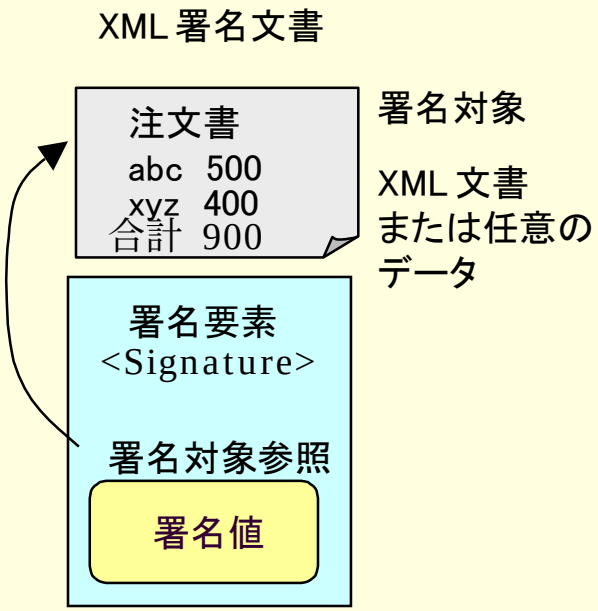
```
<Signature Id="MyFirstSignature"     ←署名要素
        xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>                                          ←1、署名情報
    <CanonicalizationMethod   ←SignedInfo要素の正規化情報
              Algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"/>
    <SignatureMethod          ←署名方法
              Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/> ←署名アルゴリズム
    <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/"> ←署名対象
      <Transforms>          ←署名対象文書の正規化方式（C14N)
        <Transform Algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026"/>
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>  ←ハッシュ値
    </Reference>
  </SignedInfo>
  <SignatureValue>MC0CFFrVLtRlk=…</SignatureValue>  ←2. 署名値
  <KeyInfo>                                          ←3. 署名検証鍵情報（オプション）
    <X509Data>  a5xgv4yrk  </X509Data>  ←X.509公開鍵証明書
  </KeyInfo>
  <Objects> ….</Objects>                             ←4. 署名関連情報（オプション）
</Signature>
```

XML署名の特徴

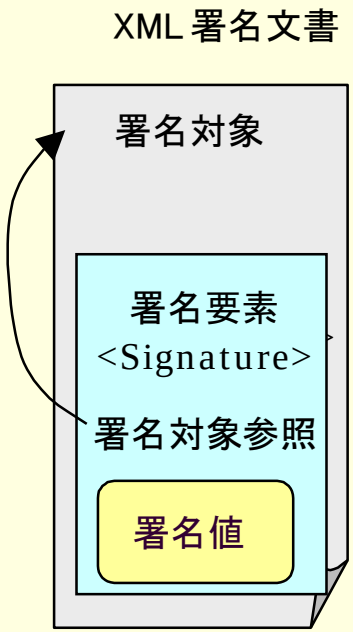
- 任意のファイルの署名（XML以外のファイルもOK)
- XMLの特定のエレメント、コンテントへの署名
- 署名検証者に鍵情報<KeyInfo>にX.509certやKerberossトークンなどを指定できる
- 証明書の有効性は検証しない（公開鍵は正しいとする→検証はXKMSなどに任せる）
- 3タイプのXML署名
  - Detached Signatures
  - Enveloping Signatures
  - Enveloped Signatures



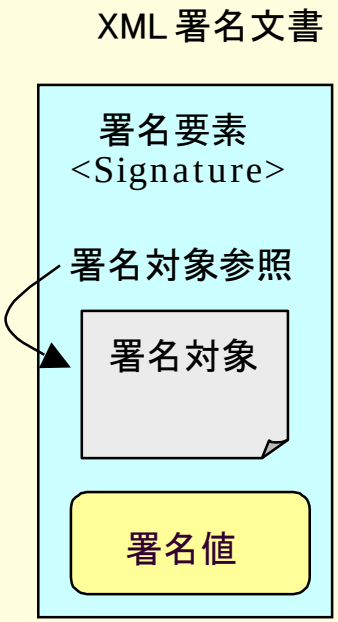
XML署名の3つのタイプ



Detached署名



Enveloped署名



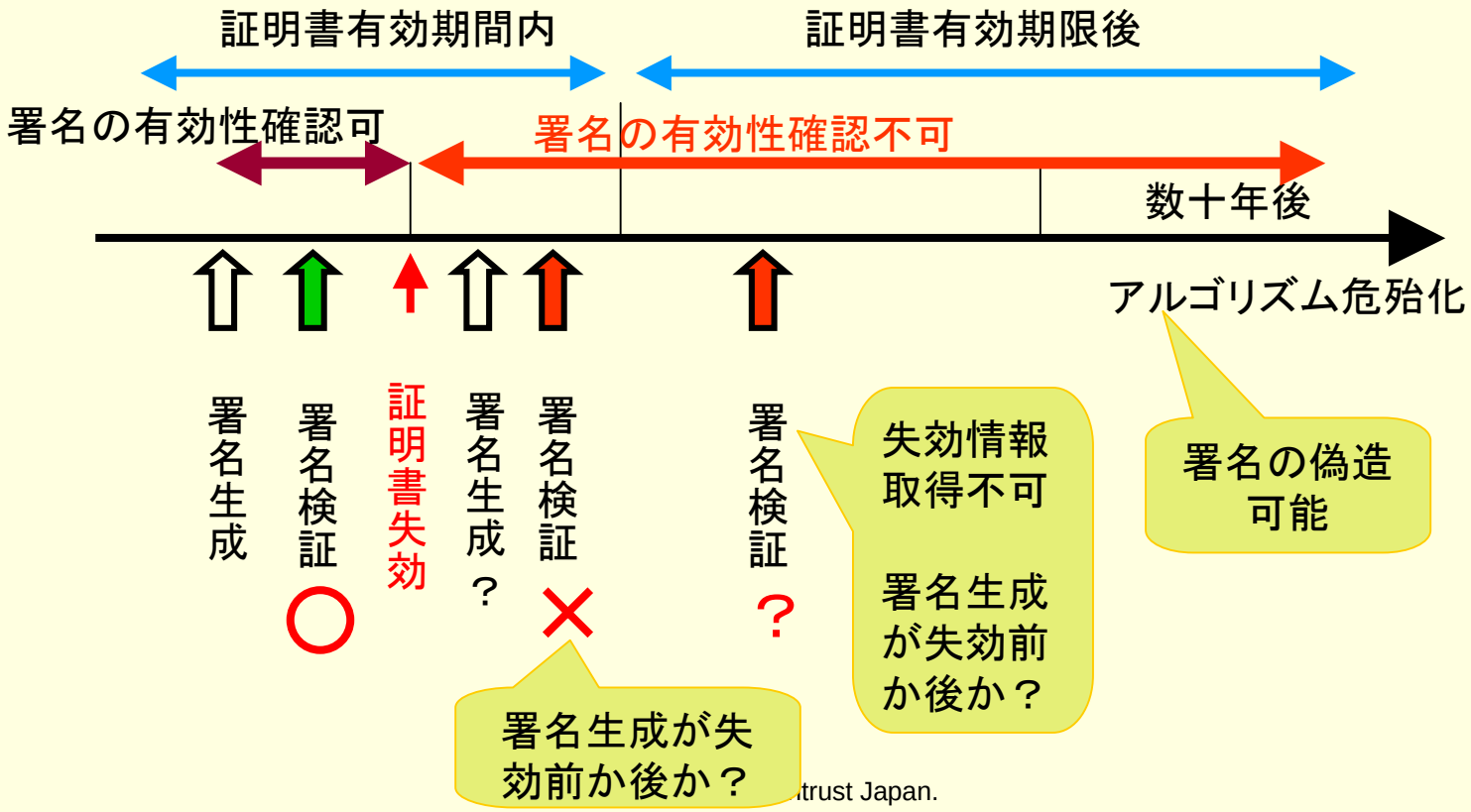
Enveloping署名

長期署名の検証の問題ー通常のPKI環境の限界

- 通常のPKI環境
  - 証明書有効期間切れると署名検証ができない！
    - 過去の鍵情報が保証されない
    - 失効情報（CRL）が利用できなくなる
    - 署名時点の信頼できる時間情報がない
    - 署名検証時点で失効されていなかったかが分からない

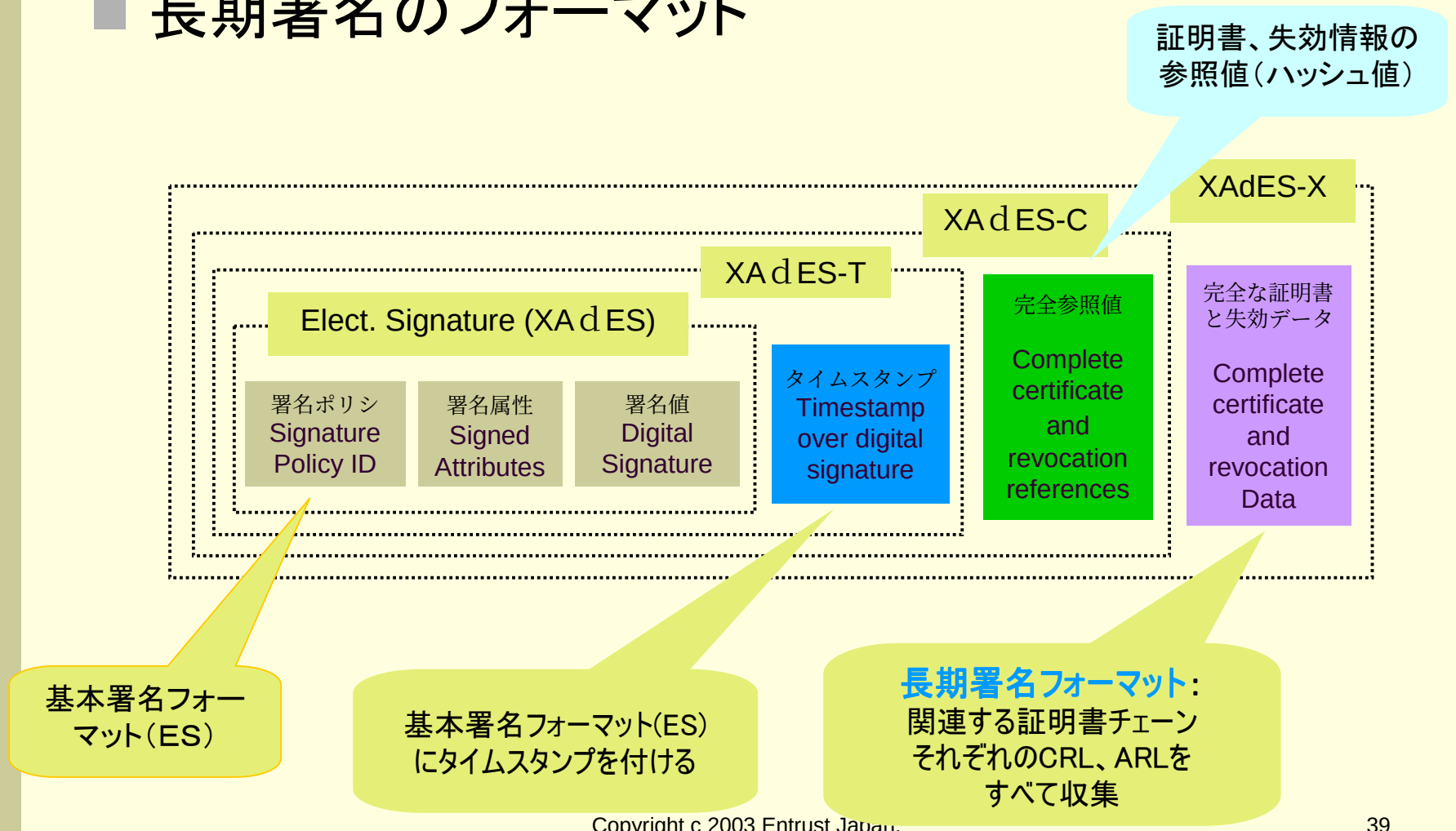
PKI環境でのデジタル署名の有効性

- 証明書失効後は、署名が失効以前に生成されたか、失効後に生成されたか分からない



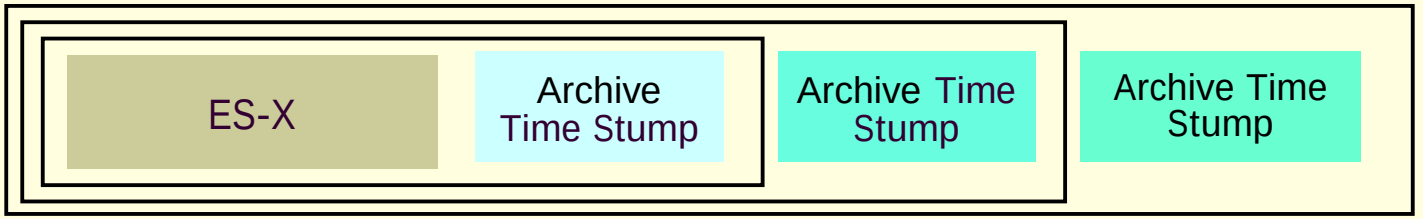
XAdES 長期署名のフォーマット

- 長期署名のフォーマット

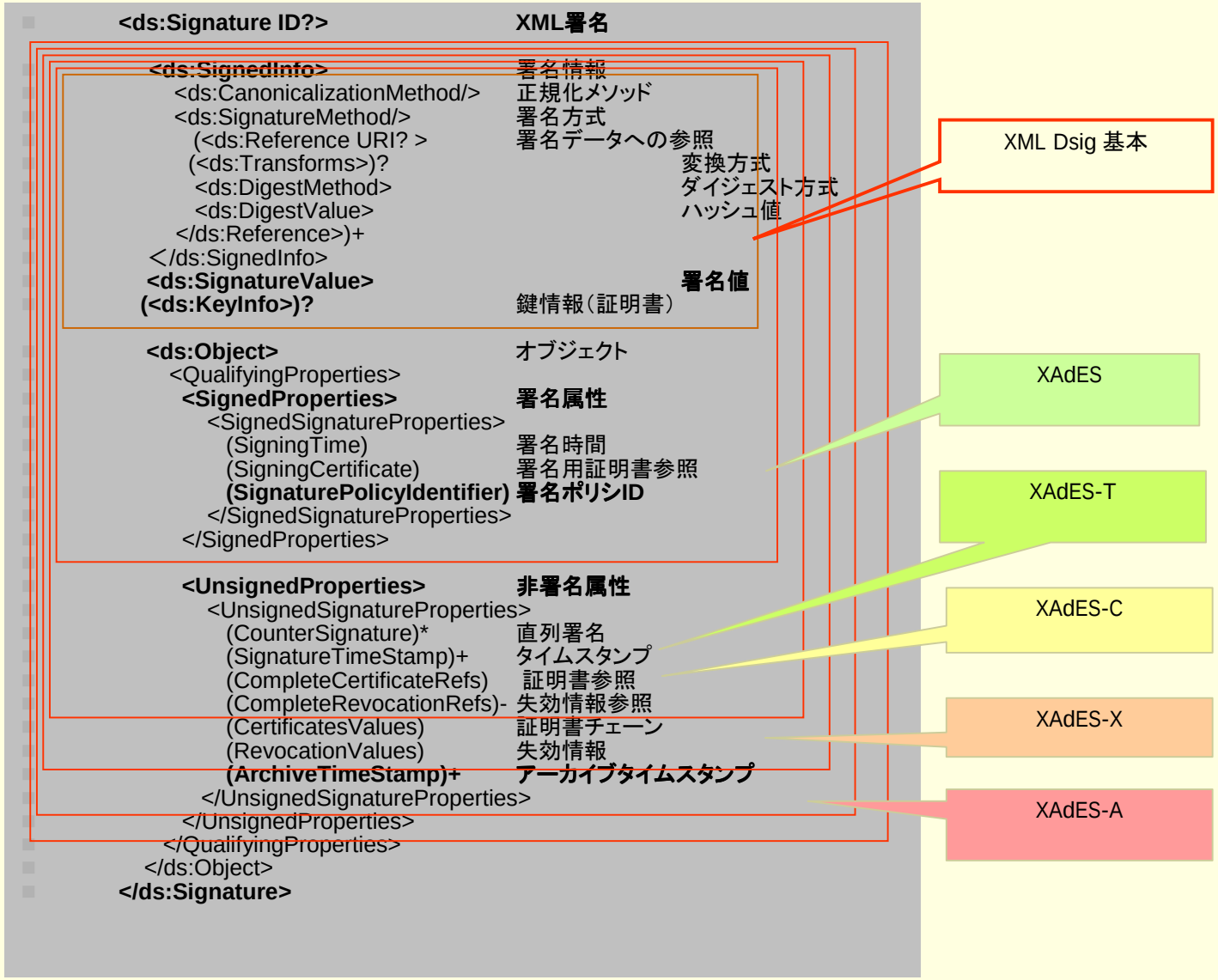


長期記録のための電子署名

- 何十年も保存する署名文書(Archive)
  - 暗号強度の弱体化、署名アルゴリズムの危殆化による署名偽造を防ぐ
- XAdES-A
  - 暗号方式が既に解読されるようになっても署名時点の有効性を検証できる方式 (ES-Xにアーカイブ・タイム・スタンプを付ける)
- XAdES-Aの延長方法
  - 前のアーカイブ・タイムスタンプの有効期限切れ前に、検証の有効性確認のため新しいアーカイブ・タイム・スタンプを付加する



XML長期署名フォーマット（XAdES)



XML暗号

- XML Encryption Syntax and Processing
  - W3C Recommendation (2002.12)
- 暗号化対象
  - 任意のファイル(非XMLも含む）の暗号化
  - 任意のXMLエレメント、コンテントの暗号化
- XML暗号プロセッサが鍵情報<KeyInfo>を基に対象部分を暗号化し、<CypherData>に置き換える
  - 対称鍵で暗号化
  - 公開鍵で対称鍵を暗号化
- XML復号プロセッサが<CypherData>を元に戻す
- 使用する鍵は正しいものとする
  - →公開鍵の有効性検証はこの仕様の外部で行う
  - →XKMSなどに任せる

XML暗号の例

指定したエレメントを暗号化（W3C:標準化の最終段階）

XML暗号化の例（人事データ）

```
<?xml version="1.0"?>
<employee id="123456">
  <name>鈴木優一</name>
  <title>CTO</title>
  <salary>
    <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
     Type="NodeList">
      <CypherData><CipherValue> AbCd..XyZ </CipherValue>
      </CypherData>
    </EncryptedData>
  </salary>
</employee>
```

<salaryエレメントの内容が<EncryptedDataエレメントに置きかえられる

従業員ID：123456
氏名： 鈴木優一
タイトル： CTO
給与： AbCd..XyZ

必要な部分のみを暗号化

暗号化された値

XML署名、暗号Toolkit

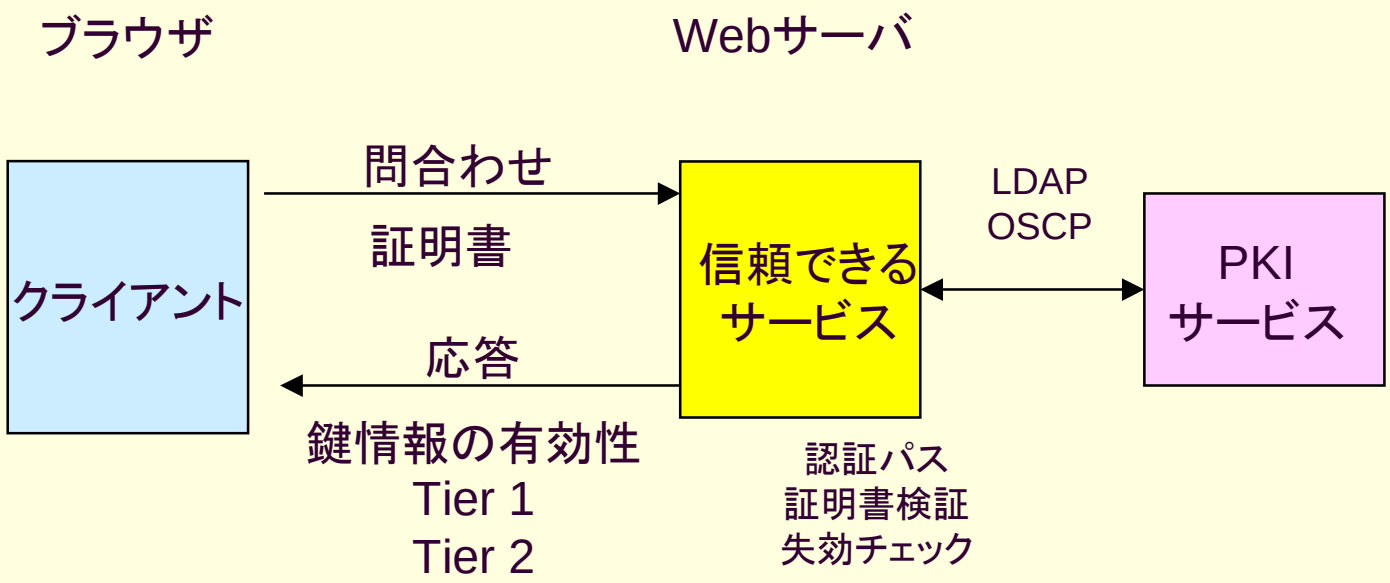
- **各社のToolkits, SDKがある**
  - Baltimore
  - DataPower
  - Entrust/Toolkit™ ; for Java™
  - IAIK XML Signature Library (IXSIL)
  - IBM XML Security Suite
  - Infomosaic
  - Microsoft
  - NEC XMLDSIG
  - Phaos
  - RSA BSAFE Cert-J
  - Ubisecure
  - VeriSign
- **Entrust Security Toolkit for Java (6.1)**
  - 簡単なJavaプログラムでXML署名、XML暗号を生成
  - XML Digital Signature処理
  - 3タイプのXML署名サポート
  - XML Encryption処理サポート
  - 署名検証、暗号復号処理もサポート

XKMS　XML鍵管理

- XML Key Management Specification 2.0
  - W3C Draft Last Call (2002.4)
- Webベースの鍵管理サービス
- 2つのサービス（要求と応答）
  - 鍵情報サービス（**X-KISS** :Key Information Service）
    - XML署名，暗号の鍵情報に関する問合わせと応答
    - 鍵情報の有効性問合わせ（証明書検証）
  - 鍵登録サービス（**X-KRSS** :Key Registration Service ）
    - 鍵の登録プロトコル（証明書発行要求）
    - バックエンドのPKIとのリンケージ
- クライアント･サーバメッセージングはSOAPに載せる
- http://www.w3.org/TR/xkms/

XKMSサービス

X-KISS（鍵情報の有効性検証）



XKMS　X-KISS（鍵情報サービス）

- クライアントがXML署名の鍵情報についての処理をXKMSサーバに委任（公開鍵証明書の有効性検証など）
- <ds:KeyInfo>エレメントについての問合わせ応答
  - X.509証明書の構文解析
  - ディレクトリから証明書、CRLを取得
  - 失効情報の検証
  - 認証パス構築、検証
- Tierサービスモデル
  - Tier 0 : PKIなどTrustサービスを用いないで鍵情報を得る
  - Tier 1 : 鍵所在情報サービス (Locale)
  - Tier 2 : 有効性検証サービス (Validate)

XKMS　X-KRSS（鍵登録サービス）

- **X-KRSS**
  - ユーザ毎の公開鍵登録のプロトコル
  - 鍵ペアをクライアント生成の場合
  - 鍵ペアをサーバで生成の場合
  - PoP確認
  - 鍵失効要求
  - 鍵回復サービス
  - バッチ鍵登録サービス
    - ICカードのバルク発行

XMLセキュリティの基本デモ（XML署名と署名検証）

- XML署名生成
  - 署名Toolの動作
- XML署名の検証
  - 証明書の検証
  - 改ざんの検出

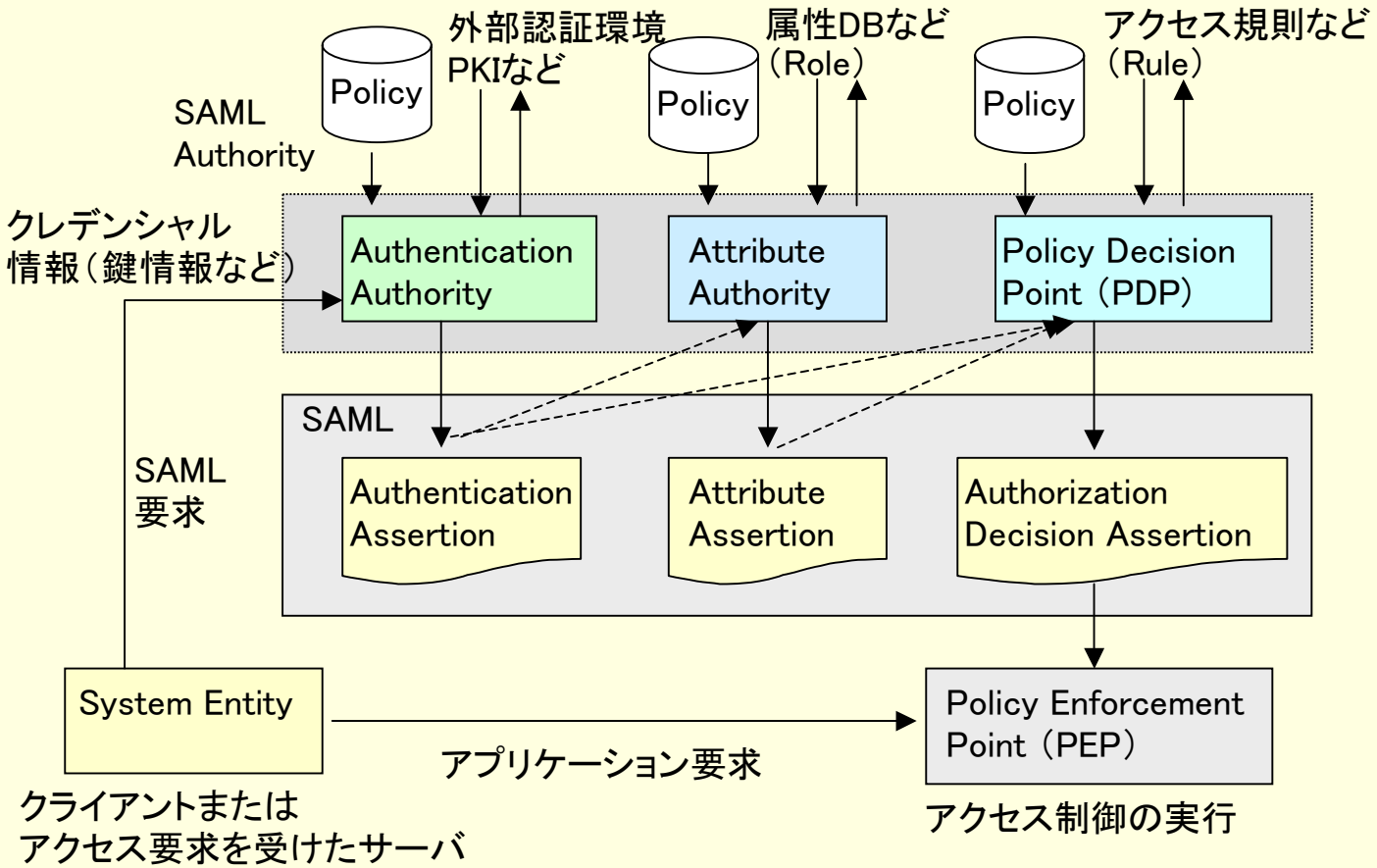
- 伊藤　康宏
- エントラストジャパン

Webサービスのセキュリティの応用

- SAML（セキュリティ・アサーション）
- Liberty（連携SSO)
- XACML（ポリシー記述）
- DSS（署名生成、検証、タイムスタンプ・サービス）
- WSS（Webサービス経路のセキュリティ）

SAML（ Security Assertion Markup Language ）

- SAML 1.1 OASIS Standard (2003.9)
- 認証、アクセス制御、属性情報の伝達



SAML Assertion

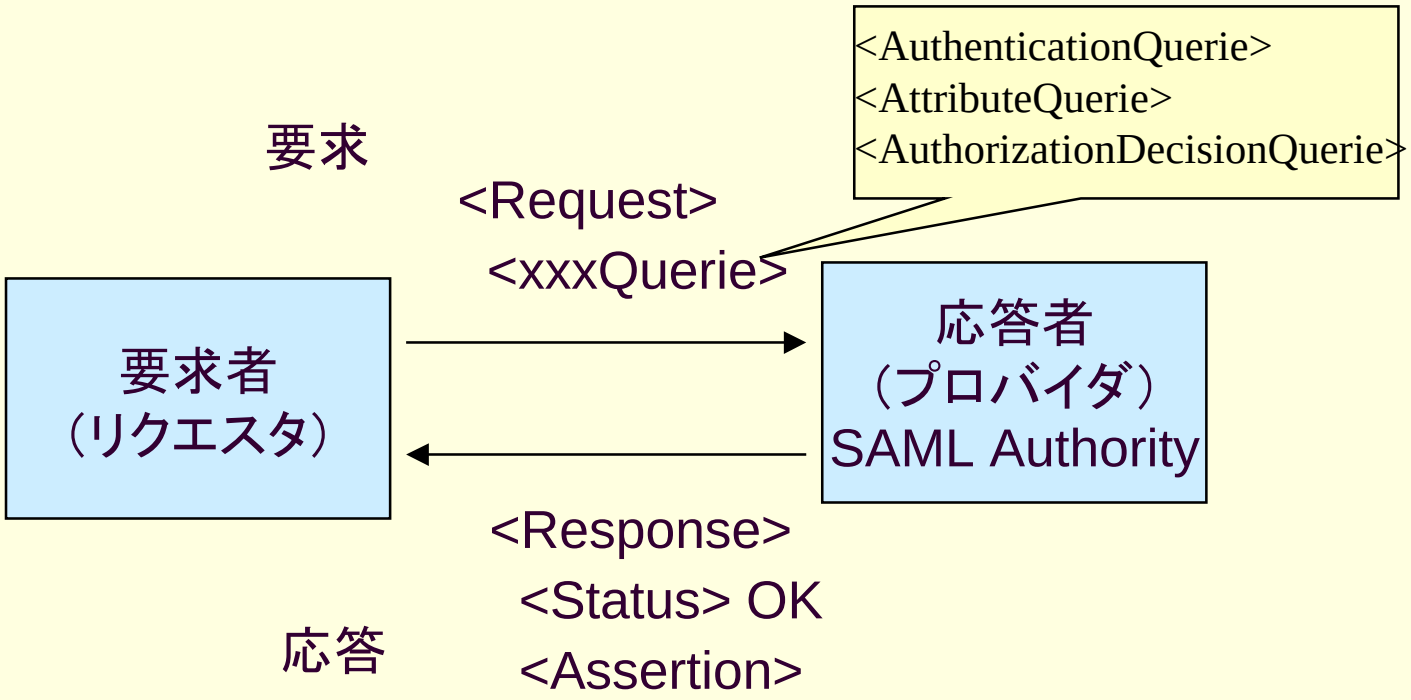
- SAML Authorityによる認証、属性、認可決定の証明

- <Assertion>
  - Assretion属性
    - バージョン番号　:1.0
    - AssertionID　:
    - Issuer　:Assertionの発行者
    - IssueInstant　:Assertion発行時間
  - <AuthenticationStatement>　:認証Statement
  - <AttributeStatement>　:属性Statement
  - <AuthorizationDecisionStatement>　:認可決定Statement

いずれか1つのステートメント

SAMLプロトコル（要求，応答プロトコル）

- 要求、応答プロトコル
  - SOAPにエンベロープ



SAMLアサーションと公開鍵証明書

- SAMLアサーション
  - 要求事項に対する証明
  - 単目的（1回のセッションで消費）
  - 短寿命（数分～数時間）
- 公開鍵証明書、X.509属性証明書
  - 公開鍵証明書：Subjectと公開鍵を結合
  - X.509属性証明書： Subjectと属性の結合
  - 多目的
  - 長寿命

SAMLアサーションポリシとCP/CPS

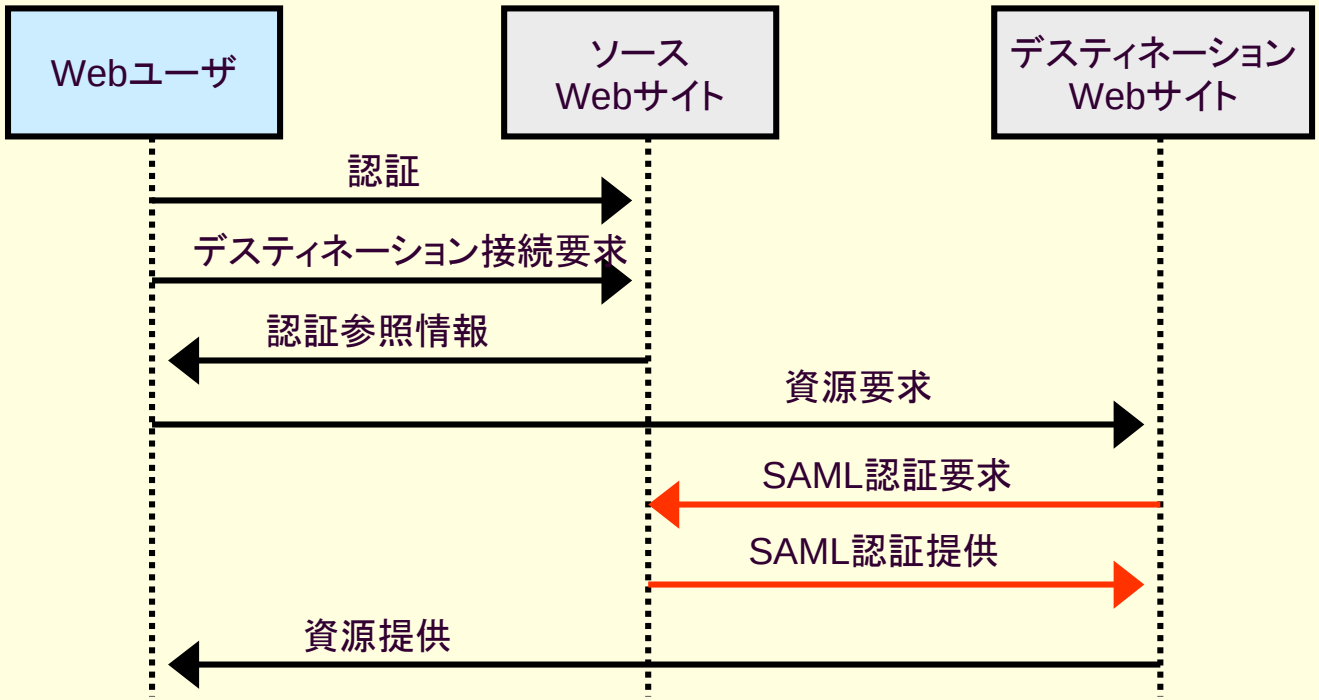
- CP/CPS:　PKIでのポリシ規定
  - CP:　証明書ポリシ
  - CPS:　認証実施規定
- SAMLでのポリシ規定
  - SAML AuthenticationMethod属性 URL
    - セキュリティの基準、ポリシ規定
  - SAML Condition要素
    - SAML Authorityの責任範囲を規定
  - SAML Advice要素
    - アサーション消費者への注意
  - SAML SubjectConfirmation要素
    - 認証方法（パスワードかPKIか）

SAMLのセキュリティ

- SAMLは特定のセキュリティ技術は規定しない
  - パスワード、Kerberos、PKI
- ピアtoピア環境
  - SSLサーバ認証とセッションの暗号化で十分
  - アサーションにデジタル署名を必要としない
- デジタル署名が必要な場合
  - <Request>、<Response>、<Assertion>にデジタル署名
  - メッセージがサーバー間を渡り歩く場合
  - アサーションをプッシュする場合
  - デジタル署名は**XML Enveloped 署名**

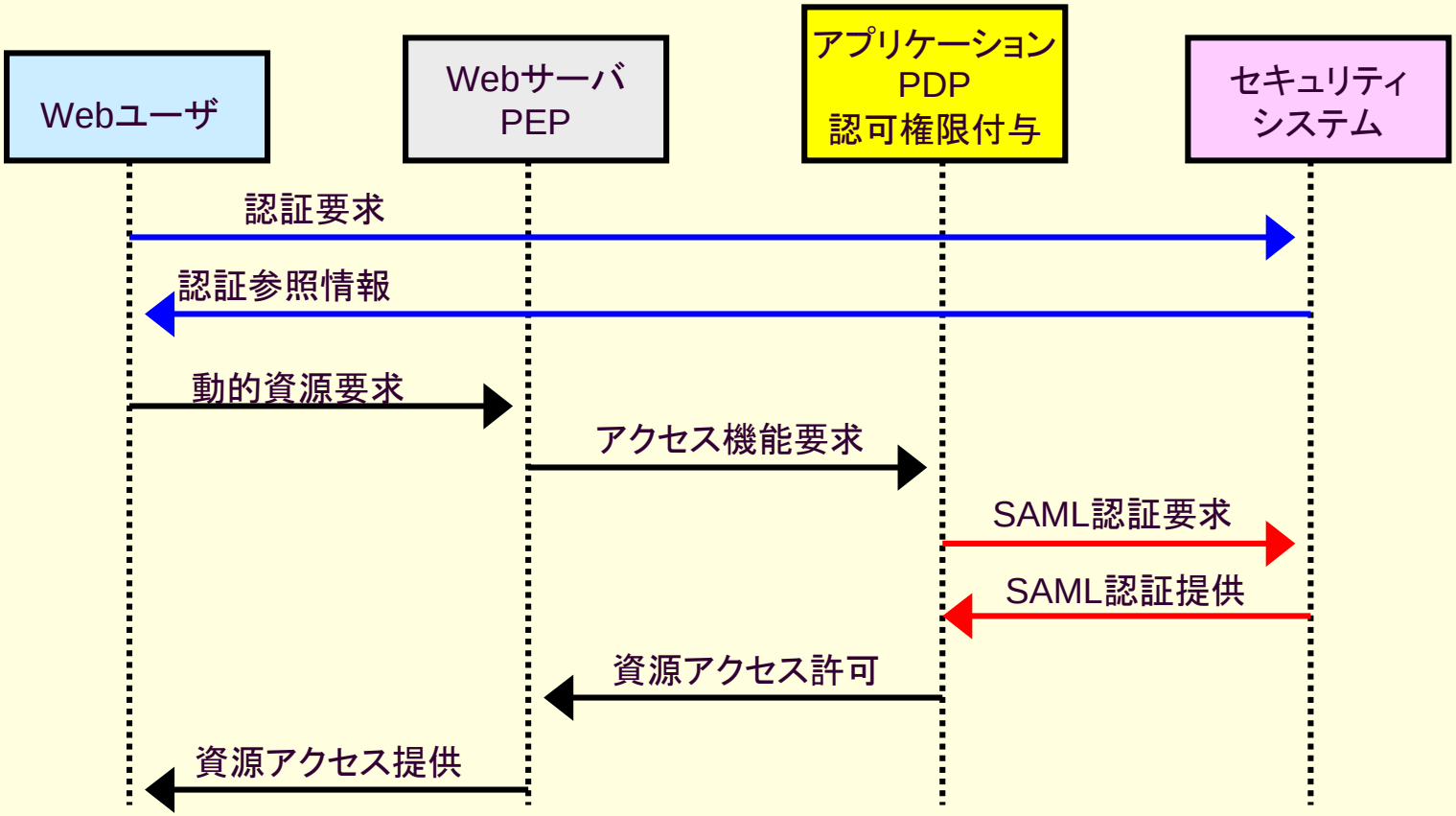
SAMLの使用法（SSO）

- SSO（Cookie認証と違いドメインを超えられる）
- SSO Pull Model



SAMLの使用法（認可サービス）

- アプリケーション・チェーン



SAML over SOAP over HTTP(要求)

- SAMLメッセージをSOAPでエンベロープしHTTPで転送（要求）

```
POST /SamlService HTTP/1.1 Host: www.example.com
Content-Type: text/xml
Content-Length: nnn
SOAPAction: http://www.oasis-open.org/committees/security
```

HTTP

```
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV=”http://schemas.xmlsoap.org/soap/envelope/”>
    <SOAP-ENV:Body>
        <samlp:Request xmlns:samlp:=”…” xmlns:saml=”…” xmlns:ds=”…”>
            <ds:Signature> … </ds:Signature>    ←XMLデジタル署名
            <samlp:AuthenticationQuery>
              … //認証問合わせ
            </samlp:AuthenticationQuery>
        </samlp:Request>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

SOAP

SAML

SAML over SOAP over HTTP(応答)

- 応答

HTTP

```
HTTP/1.1 200 OK 331
Content-Type: text/xml
Content-Length: nnnn
```

SOAP

```
<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
      <samlp:Response xmlns:samlp="…" xmlns:saml="…" xmlns:ds="…"
          StatusCode="Success">
          <saml:Assertion>
              <saml:AuthenticationStatement>
                …
              </saml:AuthenticationStatement>
              <ds:Signature> … </ds:Signature>
          </saml:Assertion>
      </Response>
    </SOAP-Env:Body>
</SOAP-ENV:Envelope>
```

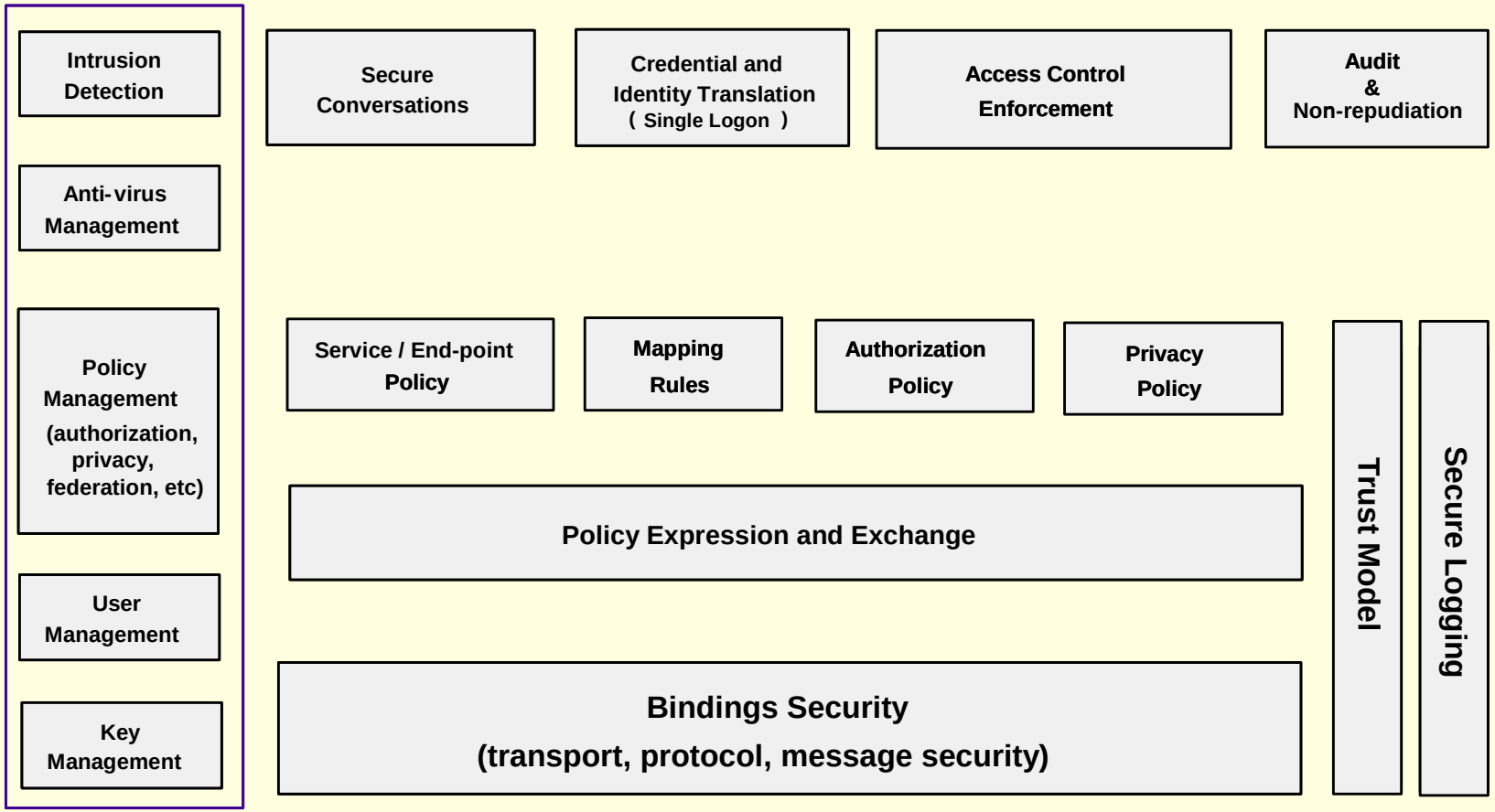
SAML

←アサーションにXMLデジタル署名

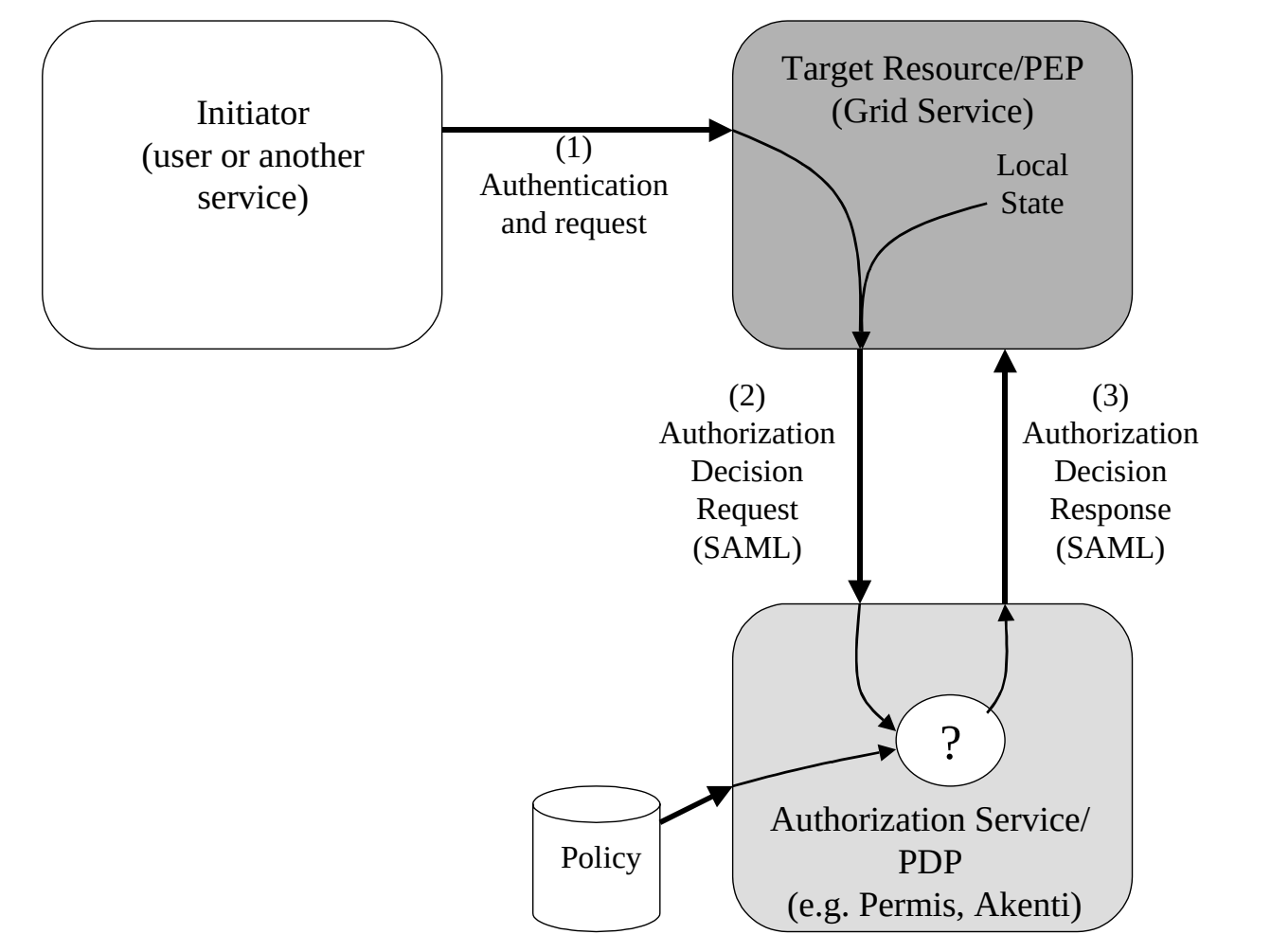
Grid Computingのセキュリティ

- Globus project
  - http://www.globus.org/
- グリッド・コンピューティングのAlliance
  - Argonne National Laboratory, the University of Southern California's Information Sciences Institute, the University of Chicago, the University of Edinburgh, and the Swedish Center for Parallel Computers,
- グリッド・コンピューティングのセキュリティ
  - セキュリティが最も重要
  - SAMLの採用

OGSA セキュリティLoad Map

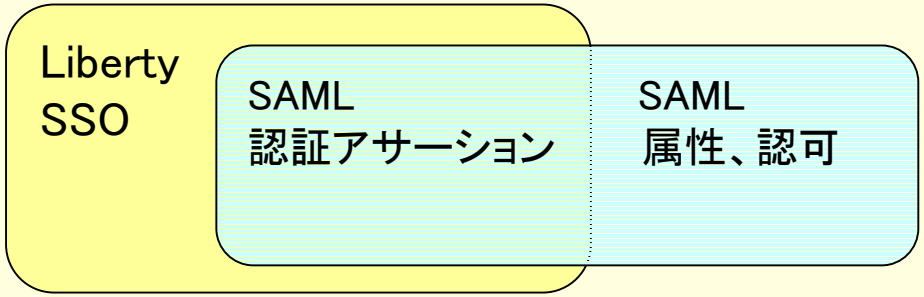


Grid Service :SAMLのセキュリティ



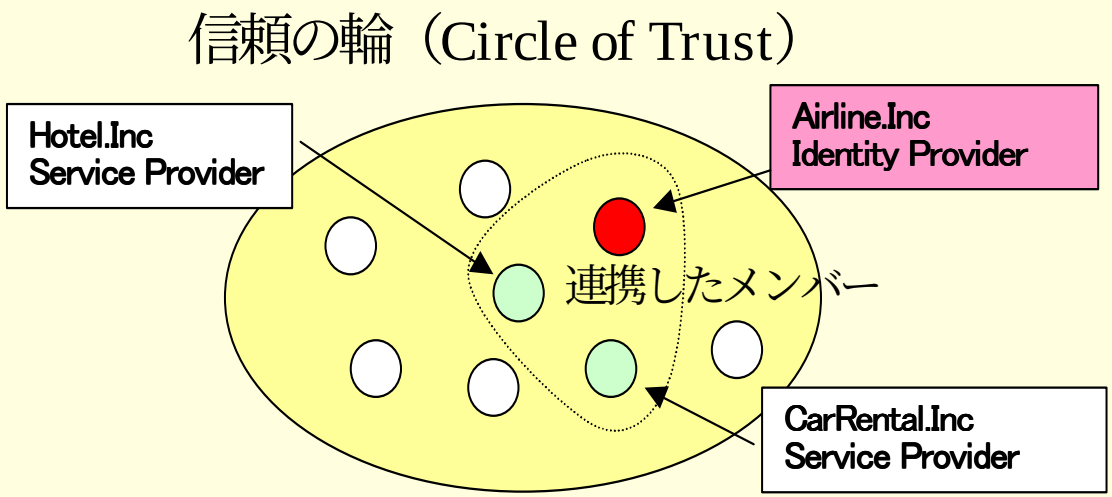
Liberty Alliance Project

- Liberty Alliance Version 1.1 Specification (2003.1)
  - 現在Phase2の作業中（2003.10）
  - http://www.projectliberty.org/
- Liberty1.1 の機能
  - マルチベンダSSOを可能にする
  - SAMLをベースにSSO機能を強化
    - 本人性連携（Federated Identity）
    - 認証（SSO）
    - グローバルログアウト
    - コンテクスト・セキュリティ
      - セキュリティ・ポリシの一致の確認



Liberty Alliance Project（Federation）

- Identity ProviderがService Providerと連携する
- メンバーの本人情報を直接交換しない(プライバシ確保)

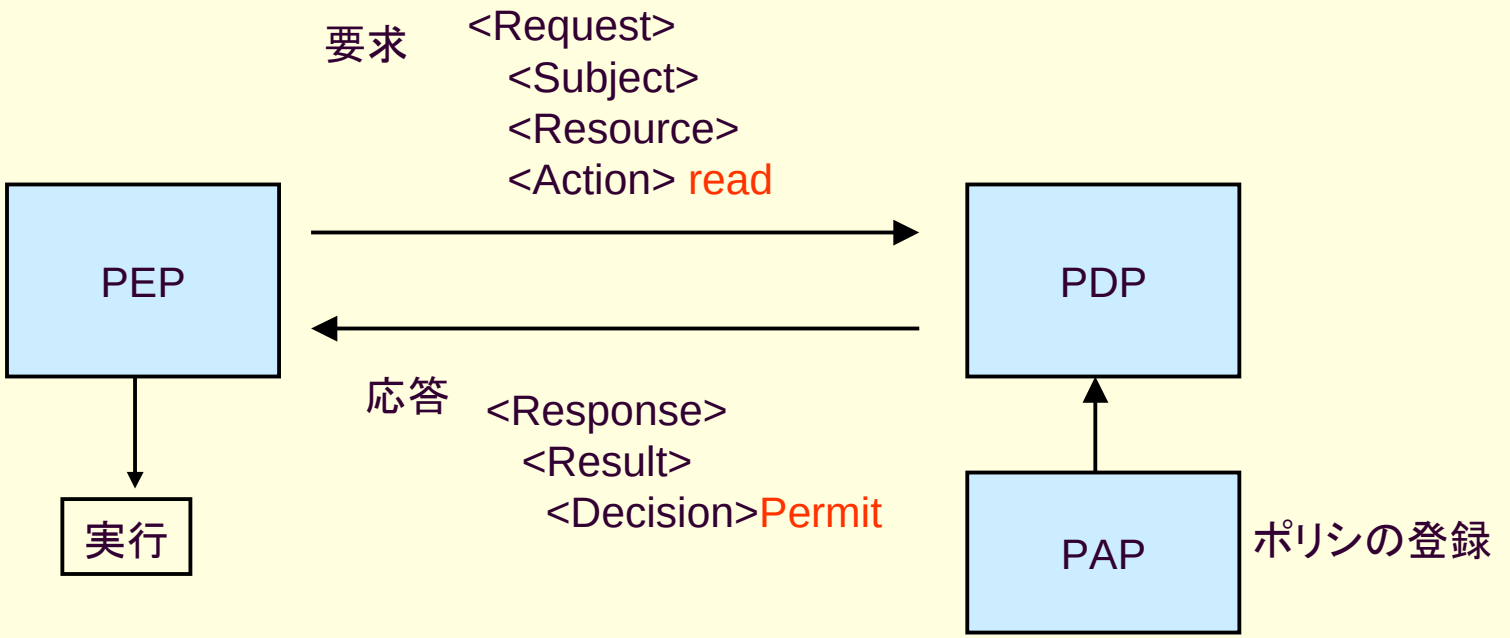


Liberty Allianceと.NET Passport

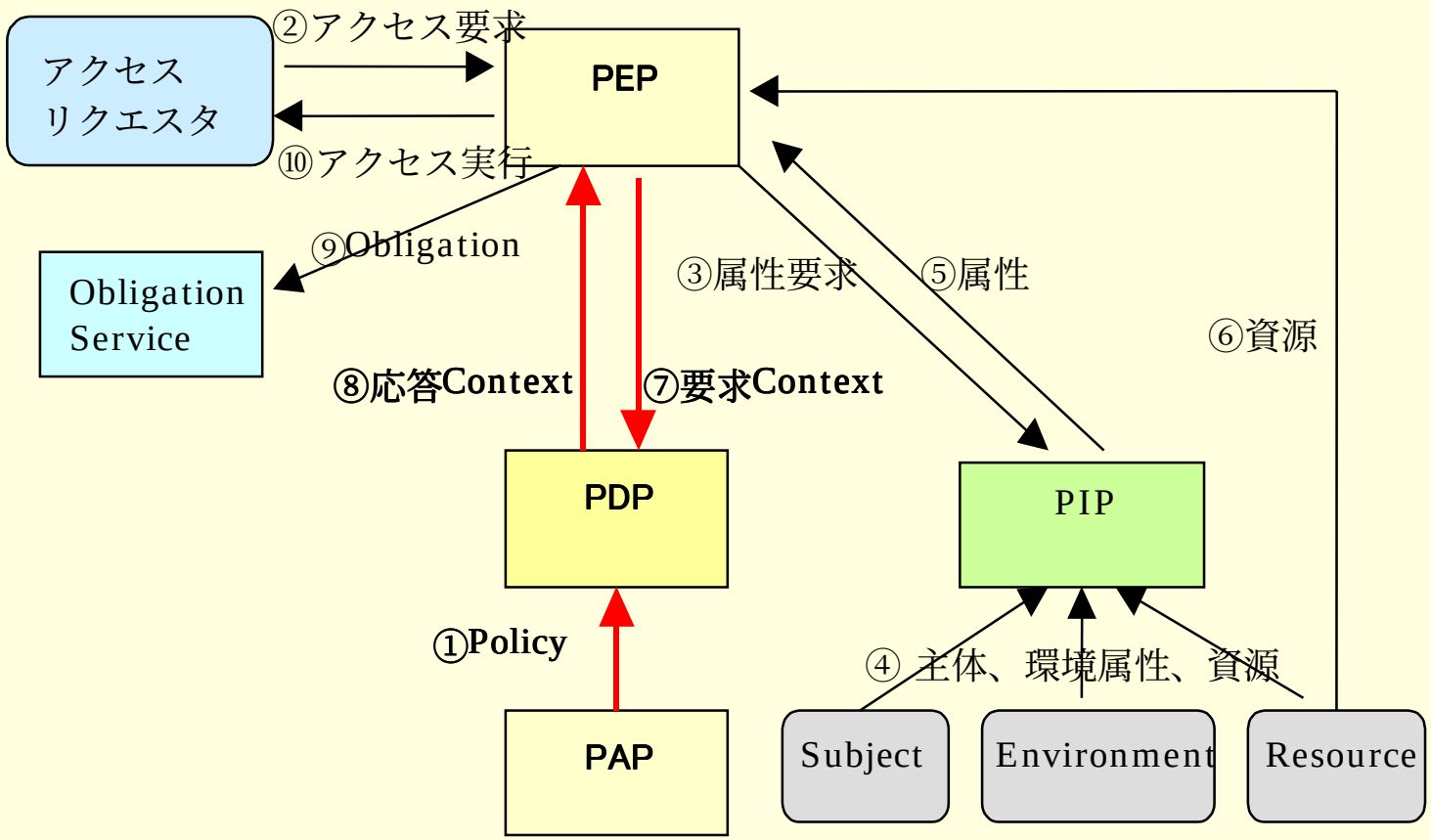
- ビジネス連携SSOへの要求
  - 企業内システムから企業間連携へ
- Liberty Alliance Project 1.1
  - SAMLベースのFederated SSO
  - SAML認証アサーションの伝達
  - PKIとの連携
  - 今後の展開が期待される（150社以上のメンバー）
- .NET Passport 2.0
  - マイクロソフト独自のサービス
  - Kerbeross 5.0ベースのSSO
  - 暗号化した認証クッキーの伝達
  - 既に多くの展開例がある
- WS Federation
  - IBM、Microsoft、RSA・・・
  - Libertyと競合

XACML

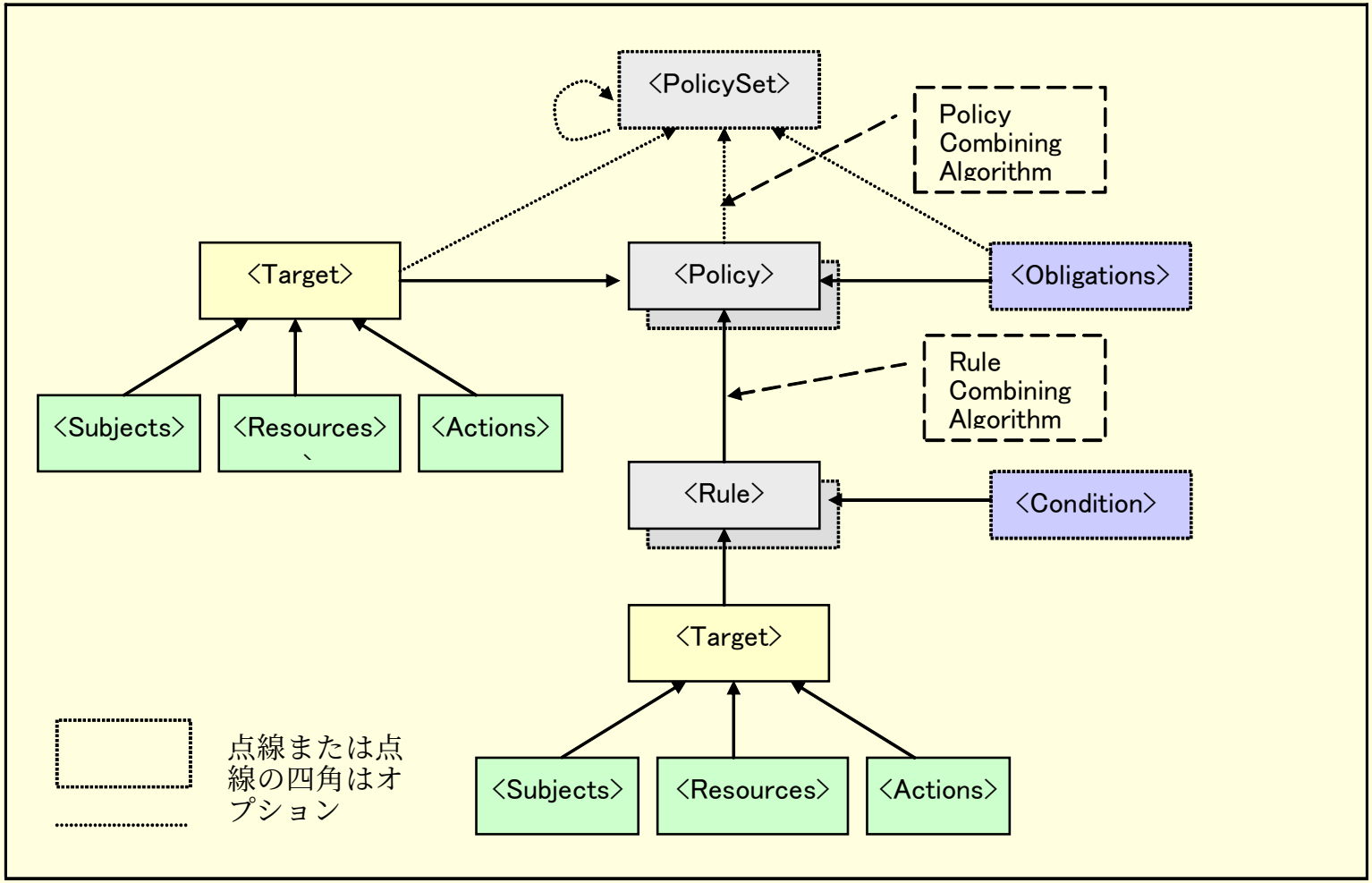
- XACML 1.0 OASIS Standard (2003.2)
- OASIS eXtensible Access Control Markup Language
- アクセス制御のポリシー記述（SAML PDPの拡張）
- トリプレット <Target> Subject＋Resource＋Action
- 要求と応答



XACMLデータフローモデル



XACMLポリシ言語構造



XACMLのPolicyの例

```
<?xml version="1.0"?>
<Policy>
  <Target>
     <Subjects><AnySubject/></Subjects>
     <Resources><AnySubject/></Resources>
     <Actions><AnyAction/></Actions>
  </Target >

  <Rule>
     <Target>
       <Subject> <uid>Suzuki</uid> </subject>
       <Resource href="creditcardnumber"/>
       <Action name="write" />
     </Target >
     <Conditions>…</Conditions>
  </Rule>
</Policy>
```

SAML, Liberty, XACML対応製品

- 多くの製品がSAML対応、または対応表明をしている
  - Entrust GetAccess 7.0
  - Netegrity SiteMinder 5.5
  - Sun ONE Identity Server 6.0
  - VeriSign Trust Services Integration Kit (TSIK)
  - Baltimore SelectAccess Version 5.1
  - RSA Security CLearTrust
  - Etc.
- マルチベンダー相互運用性
  - 今後の課題

DSS（Digital Signature Service）

- OASIS DSS TC
  - 現在策定中の作業
  - ほぼRequirementがまとまる（2003.5）
- 主な機能
  - Corporate Signature Envelope
    - 組織代表者の署名サーバ
    - 長期署名フォーマットXAdESのプロファイル
    - ビジネス文書の交換に必要
  - XML Time Stamp Protocol
    - XML署名との相性
    - TIML（Temporal Integrity Markup Language）提案ドラフト
  - Signature Validation Service
    - 最終的には署名文書の署名を検証するサービス
    - X-KISSの拡張

WSS（Web Services Security）

- OASIS WSS TC
  - 現在策定中の作業
  - IBM, Microsoft, VeriSignの仕様からOASISへ
- 主な仕様
  - Core仕様
    - SOAP Message Security Draft 11 - (3/03/2003)
      - Webサービス・ネットワークのSOAPヘッダー規約
  - プロファイル
    - SAML Token Profile Draft 6 (2/21.2003)
      - SOAPヘッダーでのSAMLアサーションの利用法
    - X509 Token Profile Draft 3
      - X.509証明書の扱い

XML対ASN.1

- IETF PKIX、S/MIME標準はASN.1ベース
  - ASN.1は効率的なコード化が可能
  - エンコードしたデータはバイナリ→可読性がない！
  - ASN.1署名標準フォーマットはCMS SignedData
  - しかし、ASN.1の理解者が少ない
- XMLでASN.1と同一構文と意味付けができる
  - XMLはマシンと人間に可読性をもつ
  - XMLの理解者は多い
  - ディジタル署名の普及にはXML署名が優れる
- PKI関連標準を全てをXMLで記述するか?
  - 基盤標準はASN.1
  - アプリケーションはXML

Q&A

鈴木　優一
suzuki@entrust.co.jp