「Webサービスセキュリティの評価、実運用時の問題点」

2002年6月13日

日本電気株式会社
NECソリューションズ　インターネット基盤開発本部
杉山　高弘(t-sugiyama@da.jp.nec.com)

本日の発表のポイント

電子政府や電子商取引における申請･調達業務において、電子文書の改ざん防止･原本を保証するために、電子署名技術の確立が望まれてます。本日は、Webサービスの電子署名運用時に必ず懸案となる以下の問題点について報告いたします。

- W3C　XML-Signature　WGにおけるInterOpの結果報告
- XMLプロセッサの非互換に関する署名検証問題
- 日本語文字コードに起因する署名検証問題
- WebサービスセキュリティのApache運用時の署名検証問題

Ⅰ.背景

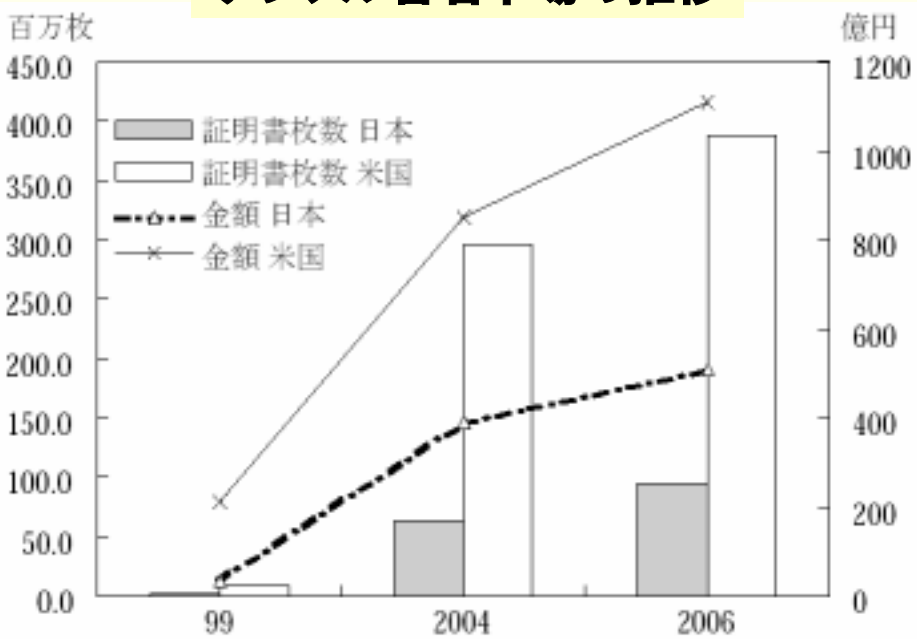
ディジタル署名の市場

デジタル署名市場は、
２００４年までに、
国内400億円規模と予想
（経済産業省）

◆電子政府・電子自治体化の進展に伴い、個人、組織の認証や、電子文書に付けた署名の電子化が必須となってくる。

◆電子文書のＸＭＬ化も進み、ネットワーク社会のインフラとして暗号化、署名の技術へのニーズが益々高まる。

デジタル署名市場の推移



（資料）通商産業省／日本ﾌﾞｰｽﾞ・ｱﾚﾝ・ｱﾝﾄﾞ・ﾊﾐﾙﾄﾝ共同調査

XML署名・暗号をコア技術とするXMLセキュリティソリューションが必須

背景

◆背景

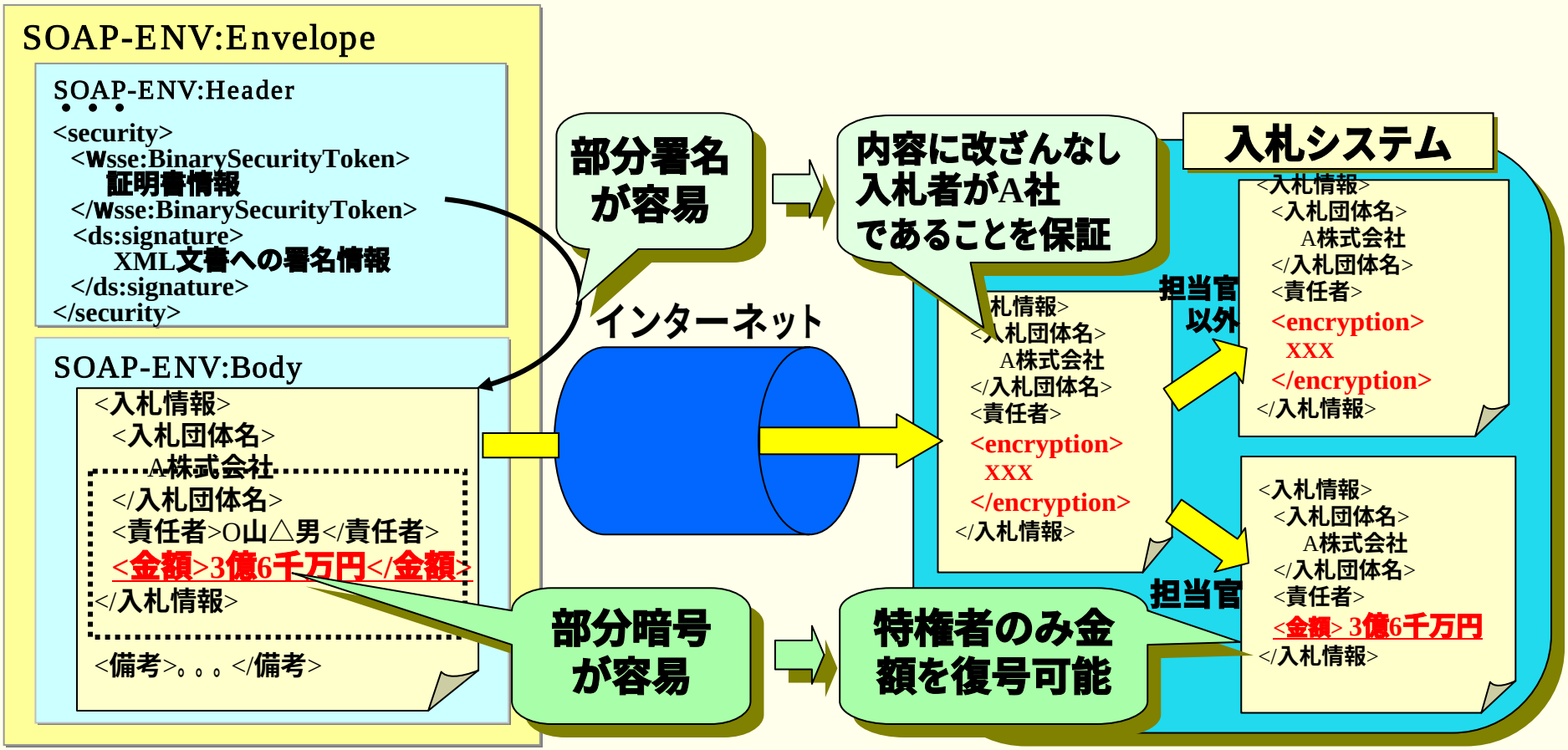
- 政府、自治体は、行政の効率化や国民負担の軽減を目標に、電子申請、電子調達システムの構築を目指している
- 電子申請文書や電子化公文書等の**本人確認、改竄防止**を実現する電子署名技術が重要

◆技術背景

- 電子文書を交換するフォーマットとして、**XMLがデファクト標準**としてますます普及
- IETF/W3C共同で“XML署名”と呼ばれる標準フォーマットが2002年2月に勧告成立。このXML署名もデジタル経済の普及とXMLの普及に伴い、**電子署名文書のデファクト標準化**が極めて有望

Webサービスセキュリティとは

電子文書の世界標準を用いた、改ざん防止、機密性確保の技術
SOAPエンベロープにおけるXML署名、XML暗号の適用
（例えば、IBM,MS, Verisign3社にて策定中の仕様）



XML署名に関する標準化、普及活動

◆W3C Signatureﾜｰｷﾝｸﾞｸﾞﾙｰﾌﾟにて標準化活動

- ➢XML署名実装ライブラリによる仕様の妥当性、互換性確認により仕様策定に貢献（互換性結果報告 NECも参加）
- ➢http://www.w3.org/Signature/2001/04/05-xmldsig-interop.html

◆XML署名リファレンスソフトウェア(NEC版)の公開

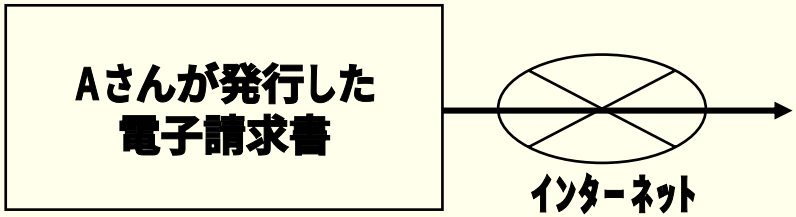
- ➢**海外公開ページ：**（XML署名ソフト（Java）、API仕様書、サンプル等）
  http://www.w3.org/Signature/　「Public Code & Tool Kits」に掲載
- ➢**国内公開ページ：**（XML署名ソフト（Java）、API仕様書、サンプル等）
  http://www.sw.nec.co.jp/soft/xml_s/appform_j.html
- ➢**ダウンロード総数：** 550件（海外）150件（国内）2002年4月時点

◆XML署名APIの標準化

- ➢XML署名コンソーシアム(NEC(幹事),日立,富士通,沖,三菱,東芝等）にてXML署名APIを策定
- ➢Java API標準化団体（Java Community Process)のXML署名標準タスクフォース(JSR-105)へ仕様案を提出

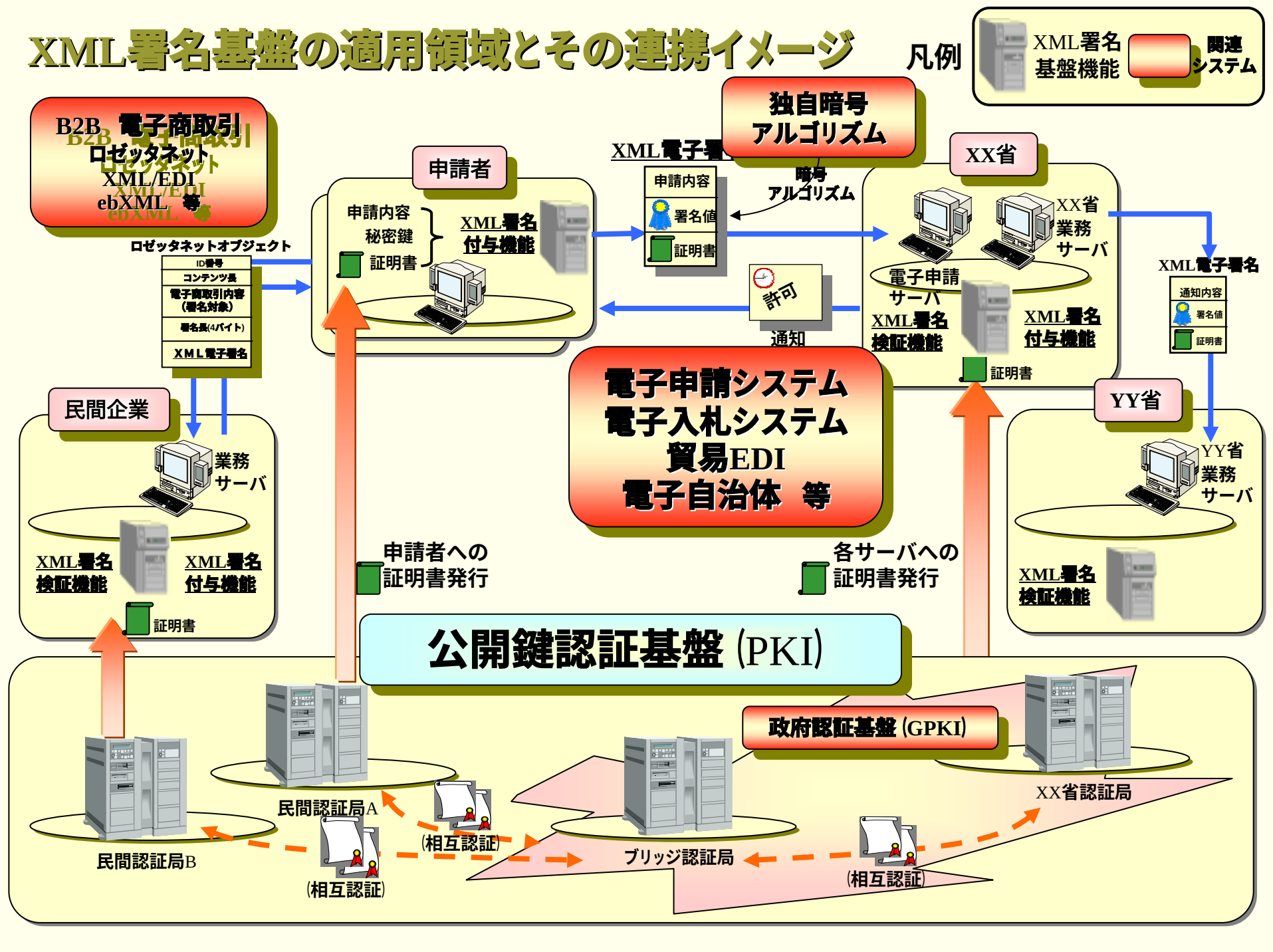
II. XML署名

電子認証･署名とは？



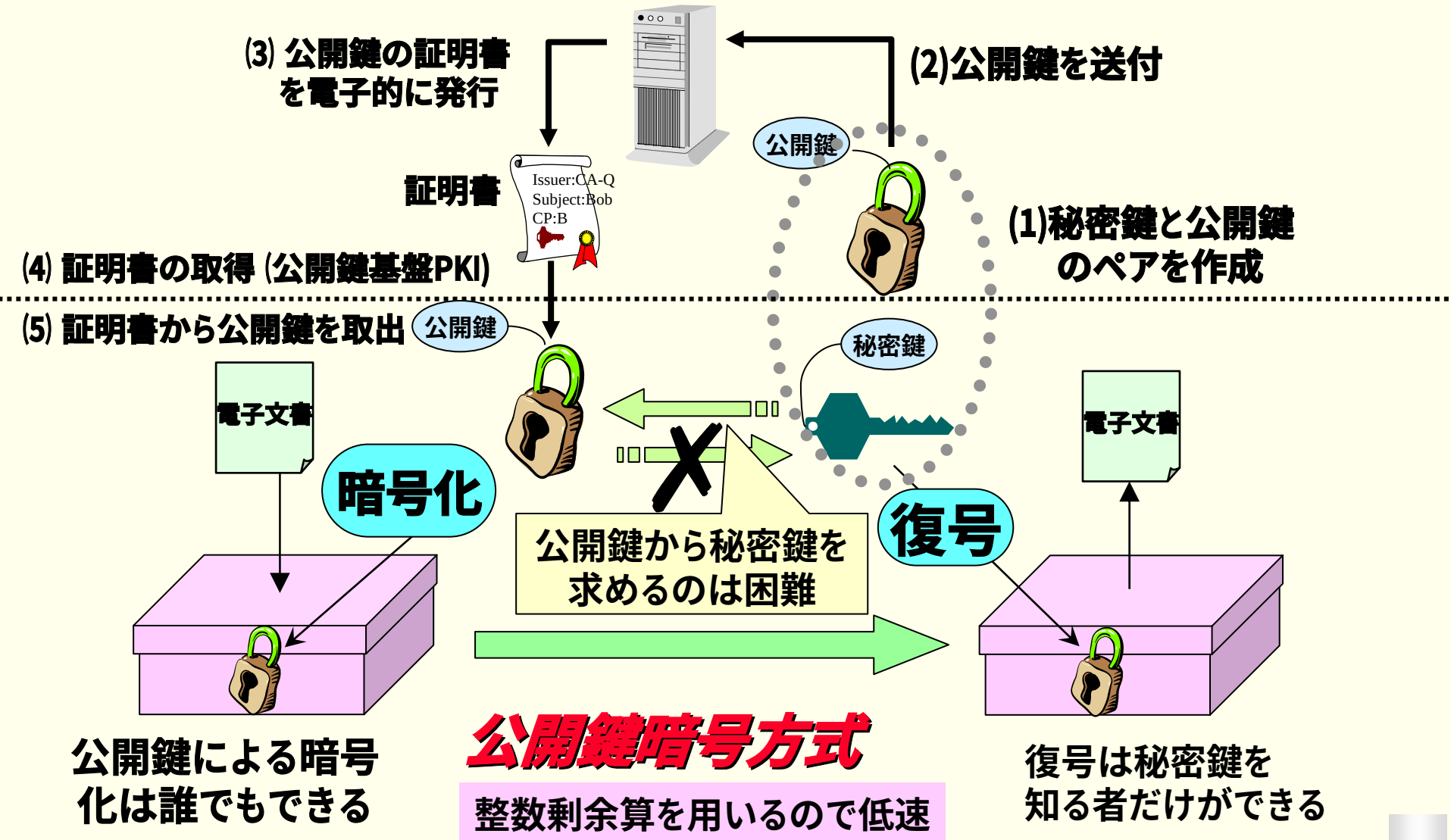
- インターネット電子取引における本人認証･改ざん防止の有効な解決手段
  ⇒ 電子署名･電子認証システム
- 電子署名　電子印鑑
  電子文書について、本人しか行い得ない操作を加えることで、その文書が本人によって適正に作成されたものであることを示す。
- 電子認証[制度]　印鑑証明[制度]
  電子認証制度は、現実世界の実印証明に相当し、当面主流となる公開鍵暗号を用いた方式では公開鍵証明書を発行する。
- 「電子署名及び認証業務に関する法律」が4/1から施行
  電子データでも手書きの署名や印鑑押印がついた文書と同程度の証拠性を持つ。(電子署名･認証技術による裏づけ)
  ⇒消費者及び企業間での電子契約が法的にも有効

XML署名基盤の適用領域とその連携イメージ



公開鍵暗号方式と認証局CA

認証局CA(Certification Authority)



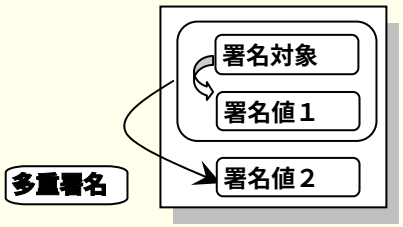
W3C/IETF　XML署名標準

- 電子署名は、電子商取引を始めとするﾈｯﾄﾜｰｸを通じた社会経済活動の安全･信頼性の確保を目的
- 標準化動向
  W3C XML Signature Syntax and Processing
  http://www.w3.org/TR/xmldsig-core/　2001/8/20にW3C勧告案（Proposed Recommendation)として成立
  Final Recommendationは、2002年冬予定
- 特長
  - セキュリティ効果
    ―文書内容の改竄防止、発信者の否認防止、成り済まし防止
  - 署名対象表現、および、署名値や証明書等をXML上で統一して表現・処理可能（プラットフォームフリー）
  - 署名対象の電子文書の種類を選ばず、部分署名や多重署名等の複雑な用件に柔軟に対応可能

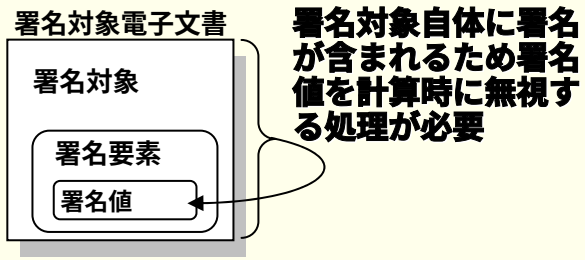
既存技術(PKCS#7等)と較べXML署名技術が優れている点

◆ 柔軟な署名モデル表現が可能

- 多重署名
  例)　Aが署名後、Bが署名
  手形の裏書の表現に使用　等

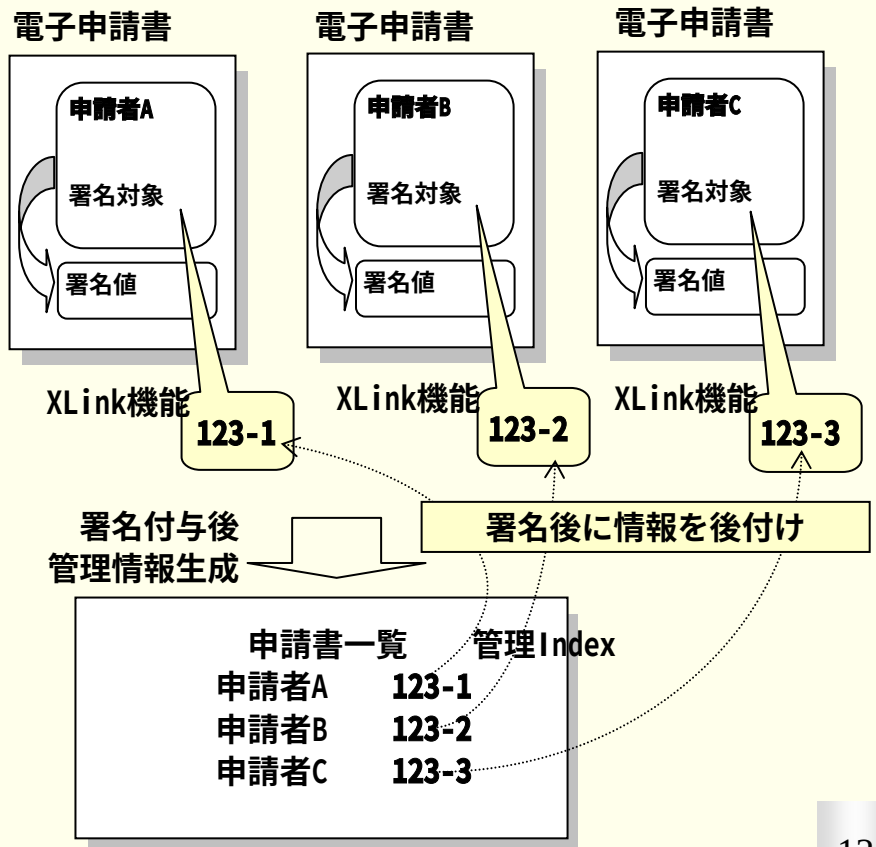


- エンベロープ形式署名
  署名対象中に署名値格納領域が内包

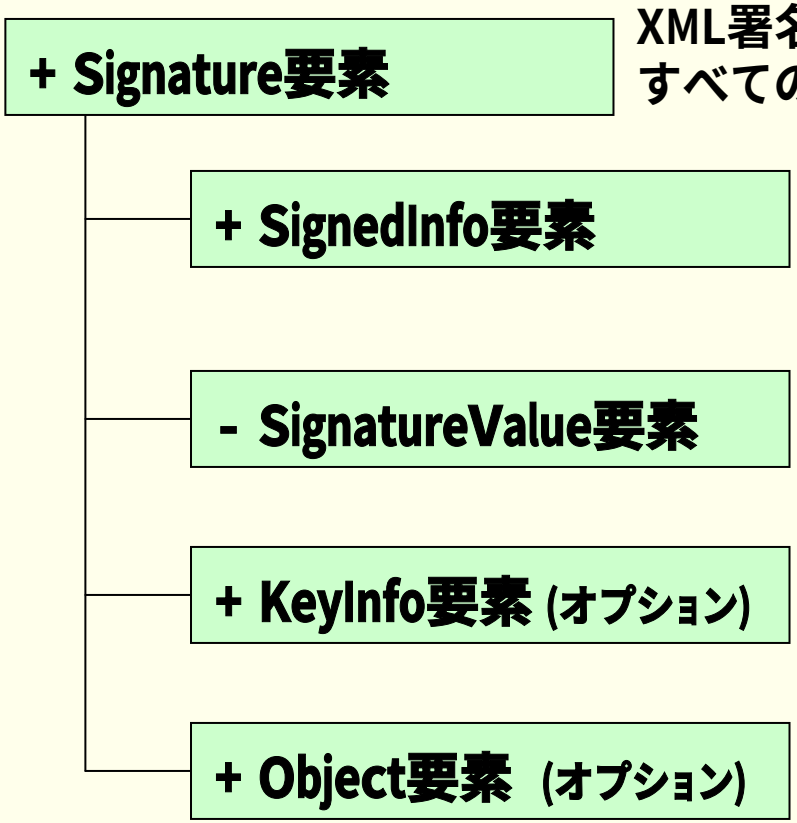


◆ 署名付与後に情報追加も可能

- 署名済み電子文書に変更を加えることなく署名後に情報を後付け可能



W3C XML署名フォーマット　構成コア要素一覧



XML署名文書のルート要素であり、XML署名に関するすべての情報を保持する。

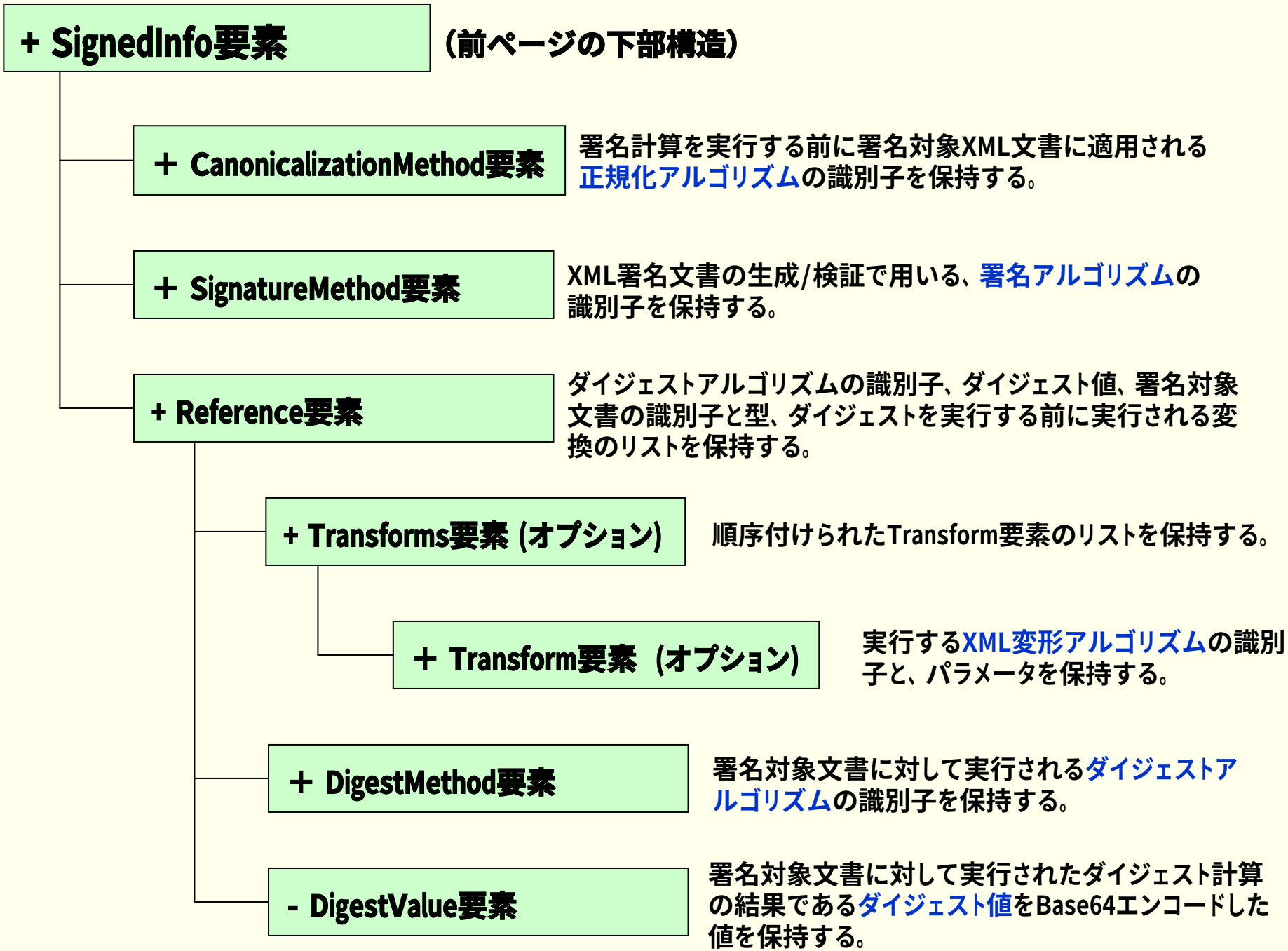
XML署名文書の生成/検証時に必要な情報（正規化アルゴリズム名、署名アルゴリズム名、および、一つ以上のダイジェスト計算関連情報　等）を保持する。

Base64エンコードされた署名値を保持する。

XML署名文書の検証で用いる公開鍵に関する情報を保持する。

アプリケーションシステムによるダイジェスト計算関連情報（Manifest要素）、署名生成時の付加的情報を保持する。

注）要素の先頭の記号の意味
　+ は、下位構造に要素を持つ
　- は、下位構造に要素を持たない


+ SignedInfo要素
（前ページの下部構造）
＋ CanonicalizationMethod要素
署名計算を実行する前に署名対象XML文書に適用される正規化アルゴリズムの識別子を保持する。
＋ SignatureMethod要素
XML署名文書の生成/検証で用いる、署名アルゴリズムの識別子を保持する。
+ Reference要素
ダイジェストアルゴリズムの識別子、ダイジェスト値、署名対象文書の識別子と型、ダイジェストを実行する前に実行される変換のリストを保持する。
+ Transforms要素 (オプション)
順序付けられたTransform要素のリストを保持する。
＋ Transform要素 (オプション)
実行するXML変形アルゴリズムの識別子と、パラメータを保持する。
＋ DigestMethod要素
署名対象文書に対して実行されるダイジェストアルゴリズムの識別子を保持する。
- DigestValue要素
署名対象文書に対して実行されたダイジェスト計算の結果であるダイジェスト値をBase64エンコードした値を保持する。

[出力データ]

1-1.xml

```
<?xml version="1.0" encoding="Shift_JIS" ?>
<Signature Id="Sig" xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026" />
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
    <Reference Id="REF_01" Type="http://www.w3c.org/2000/09/xmldsig#Object" URI="#OBJ_01">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026" />
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>VFAPKMclh+ZoZ9l3rKsxOCbWG/Y=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>GzStlmRo6FG6ogy9kIXF1/urDGJI0Epdhtk8c5Orj3Jpol...</SignatureValue>
  <KeyInfo>
    <KeyValue>
      <RSAKeyValue>
        <Modulus>ANzfAICw6ErdY7F4OR7Q+fC+h/o0AC3vbCs/w39X...</Modulus>
        <Exponent>AQAB</Exponent>
      </RSAKeyValue>
    </KeyValue>
  </KeyInfo>
  <Object Id="OBJ_01" MimeType="xml/text">
    <!-- Presentation -->
    <日付>
      <年>2001</年>
      <月>2</月>
      <日>16</日>
    </日付>
  </Object>
</Signature>
```

生成されたダイジェスト値がDigestValue要素中に埋め込まれる。

生成された署名値がSingatureValue要素中に埋め込まれる。

XML正規化(W3C Canonical XML)

- **XML正規化とは**
  - XML文書を正規化形式（Canonical form）に変換する手法
  ⇒同じ正規化表現を持つXML文書は論理的に等価
- **XML正規化の目的**
  XML処理ツールを使用する際、元のXML文書に対する変更が発生する可能性あり。
  署名対象XML文書の正規化により、論理的に等価を保障し、署名値を必ず一致させる。
- **代表的な正規化処理の例**
  (1) 非UnicodeからUnicodeへの変換
  (2) 空要素タグから開始タグと終了タグの組への変換
  (3) 属性の出現順序の整列、属性内の空白文字列の正規化
  (4) 名前空間宣言の整列
  (5) コメント部分の無視（選択可能）

XML正規化の例（１）

(1) 文字エンコーディングはUTF-8とする。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<doc>&#169;</doc>
```

XML正規化

```
<doc>#xC2#xA9</doc>
```

“コピーマーク“をUTF-8文字コードに変換

(2) 空要素は開始タグと終了タグで表し、空要素タグは使わない。

```
<quantity/>
```

XML正規化

```
<quantity></quantity>
```

(3) XML宣言およびDTDは削除する。

```
<?xml version="1.0"?>
<?xml-stylesheet href="doc.xsl” type="text/xsl"   ?>
<!DOCTYPE doc SYSTEM "doc.dtd">
<doc>Hello, world!</doc>
```

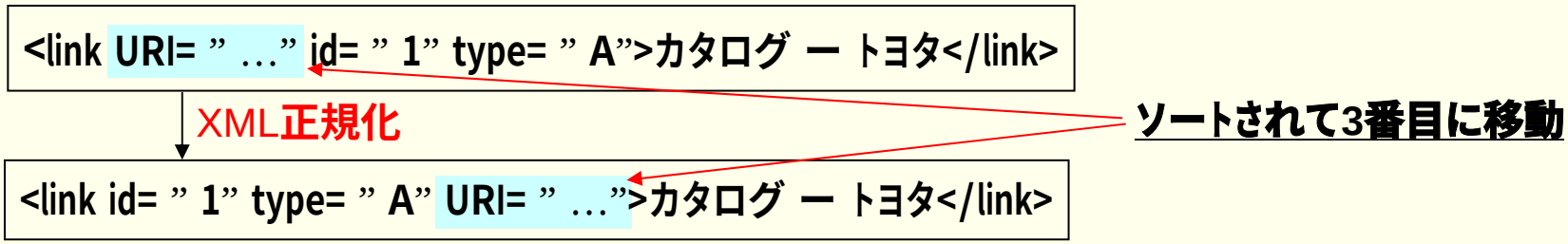
削除される

XML正規化

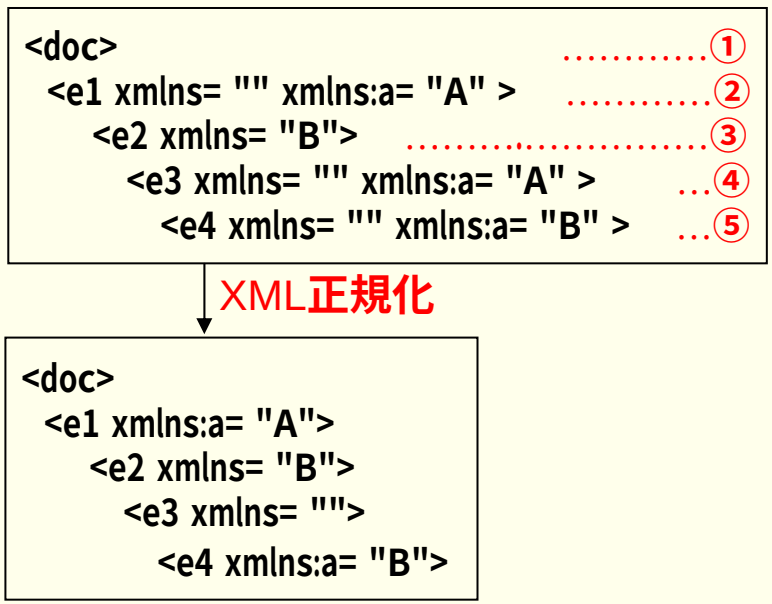
```
<?xml-stylesheet href="doc.xsl” type="text/xsl"   ?>
<doc>Hello, world!</doc>
```

XML正規化の例（２）

（4）属性は属性名の昇順にソートされる。



（5）名前空間の正規化



名前空間の正規化手順

（1）①でxmlnsは未定義のためデフォルト値 ("") に設定されている。②で""に 再設定する必要はないため、②のxmlnsの設定は不要。

（2）②のxmlns:aの設定はデフォルト値 ("") と異なるため必要。

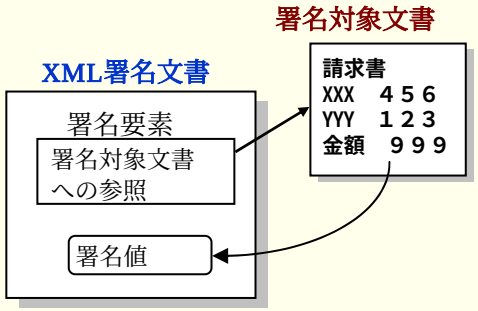
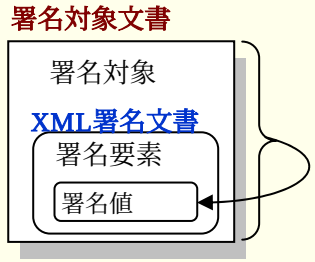
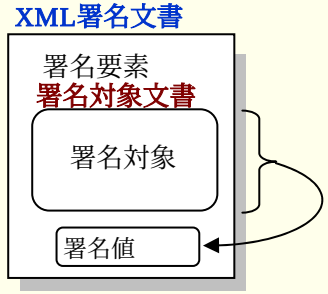
（3）②でのxmlnsの値は""であり③のxmlnsの設定は必要。

（4）③でのxmlnsの値は"B"であり④のxmlnsの設定は必要。

（5）③でのxmlns:aの値は②で設定された"A"であり④のxmlns:aの設定は必要。

（6）④でのxmlns:aの値は④で設定された"A"であり⑤のxmlns:aの設定は必要。

署名対象文書とXML署名文書の関係

種別	Detached署名	Enveloped署名	Enveloping署名
模式図			
説明	署名(Signature)要素と署名対象要素が独立な文書の署名形式	署名(Signature)要素が署名対象要素の子孫となる署名形式	署名(Signature)要素が署名対象要素の先祖となる署名形式
特徴	署名対象文書に任意の電子ファイル（Word，Excel等）を指定し、XML署名文書と独立に管理可能	現実の契約書等の署名の形式に合致した形式 EDIの電文形式相当のXMLへの署名に有効	署名対象文書と署名要素を一体として管理可能なので取り扱いが容易

Detached署名の例

(別文書の例)
署名対象要素はURIで指定。
Signature要素とは無関係。

署名対象文書が格納されている場所

署名を付与する文書

```
.....
<Signature xmlns="http://www.w3.org/2000/07/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2000/WD-xml-c14n-20001026" />
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />
    <Reference URI="http://www.nec.co.jp/hogehoge">
      <Transforms>
        <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64" />
      </Transforms>
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue> (ダイジェスト値) </DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue> (署名値) </SignatureValue>
  <KeyInfo>
    <KeyValue>
      <DSAKeyValue> (公開鍵値) </DSAKeyValue>
    </KeyValue>
  </KeyInfo>
</Signature>
.....
```

XML署名文書

署名対象文書

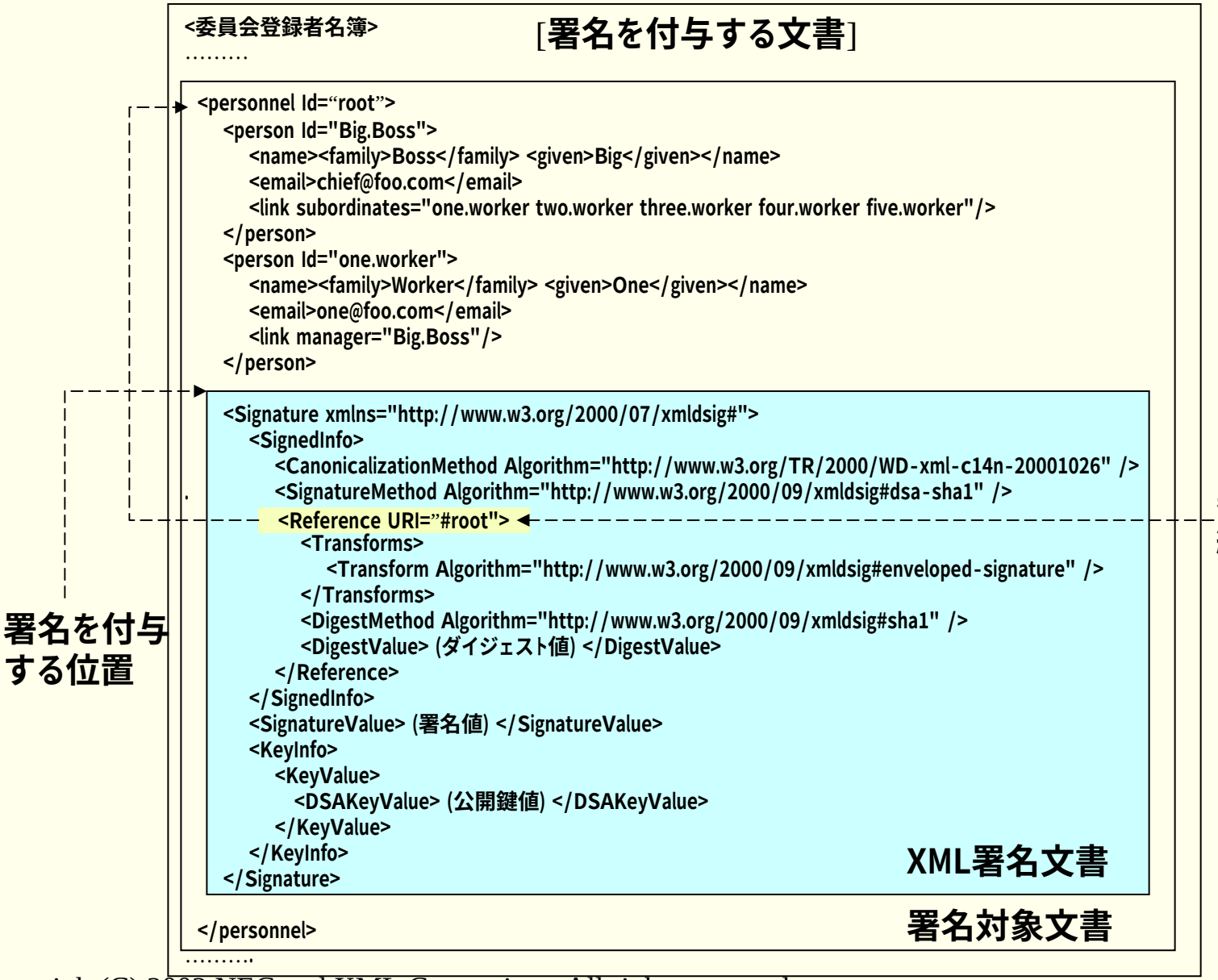
```
<personnel  Id="root">
  <person Id="Big.Boss">
  .....
```

(形式は任意)

署名を付与する位置

Enveloped署名の例

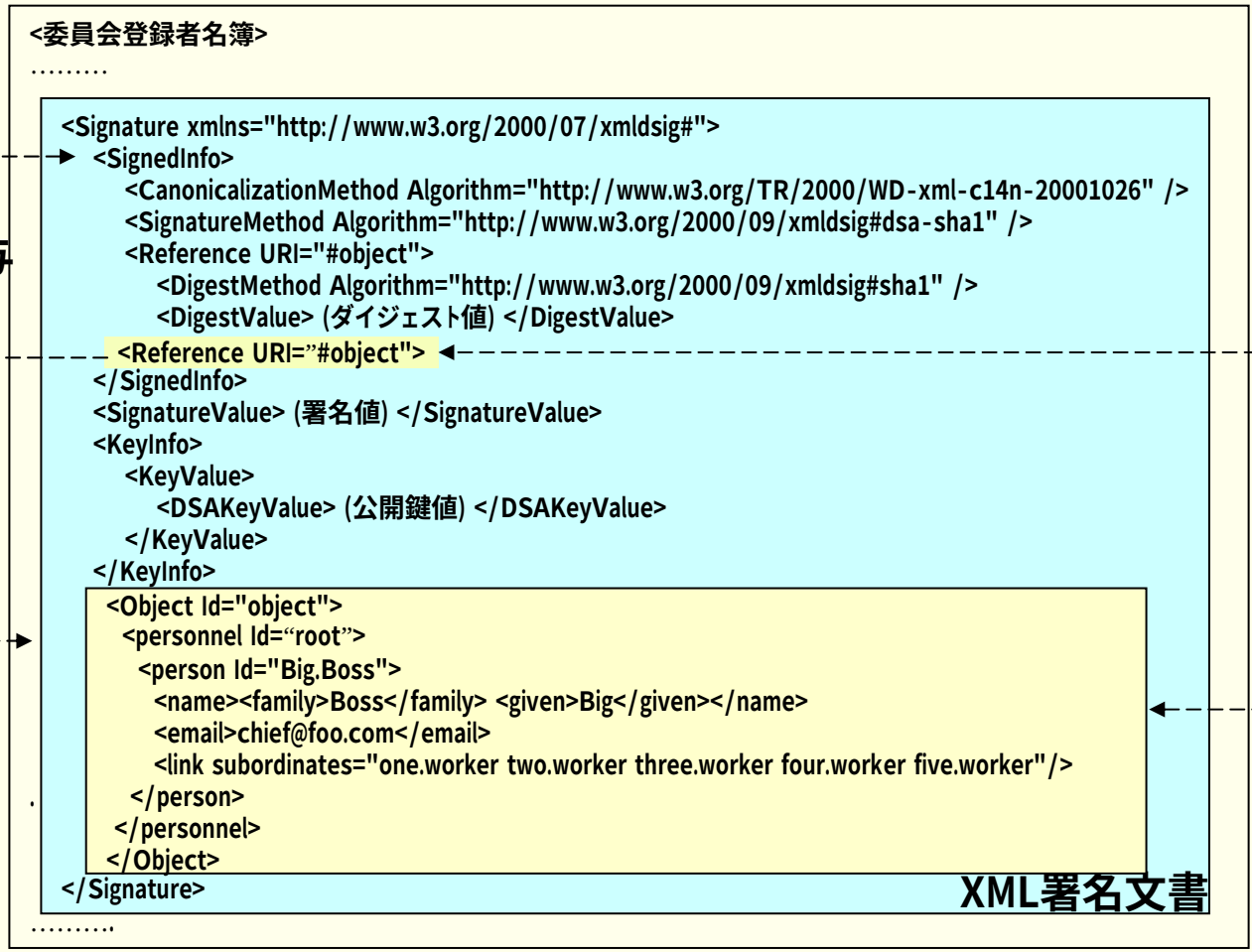
Signature要素は
署名対象文書要素の子孫



Enveloping署名の例

Signature要素は
署名対象文書要素の先祖

署名を付与する文書

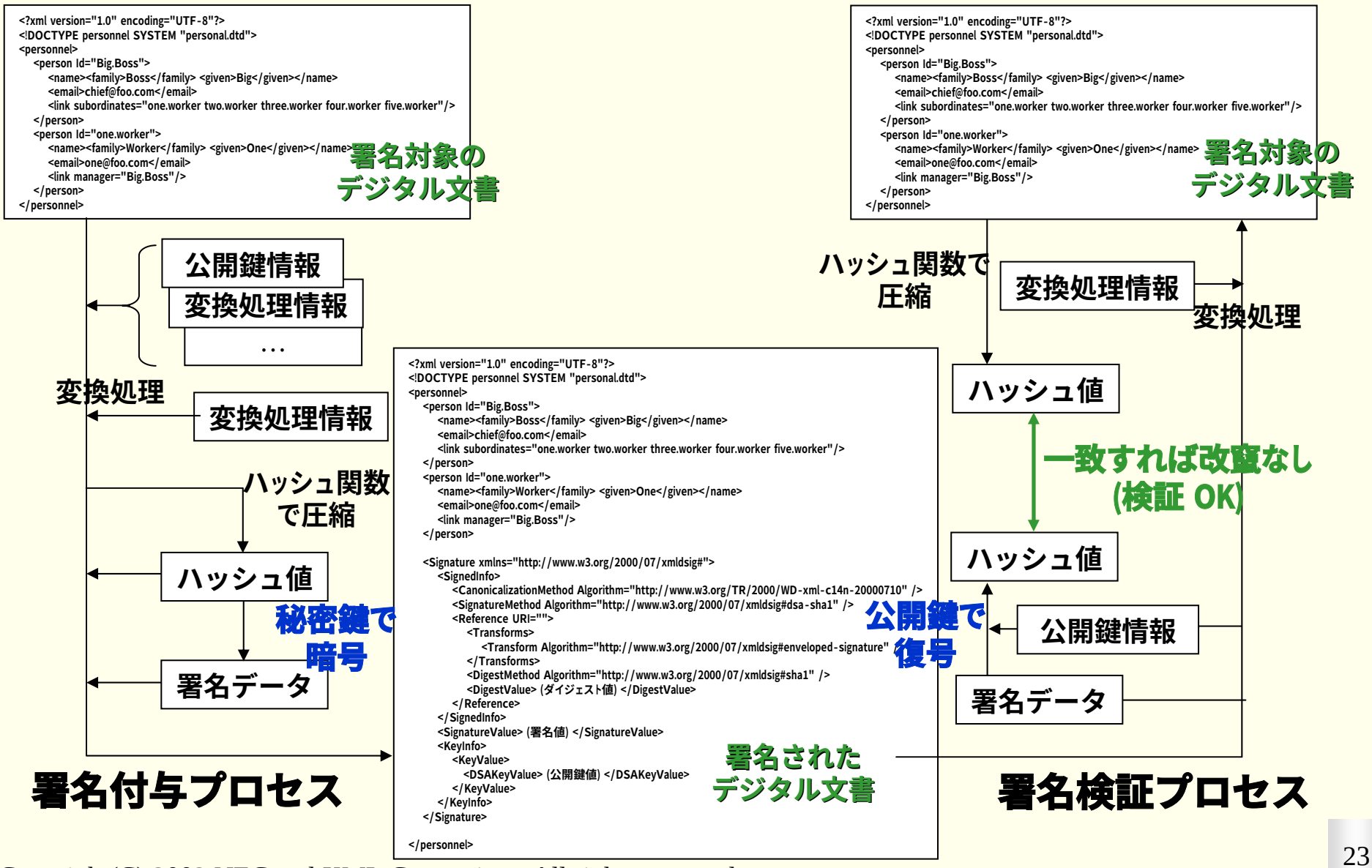


署名を付与する位置

署名対象文書が格納されている場所への参照

署名対象文書

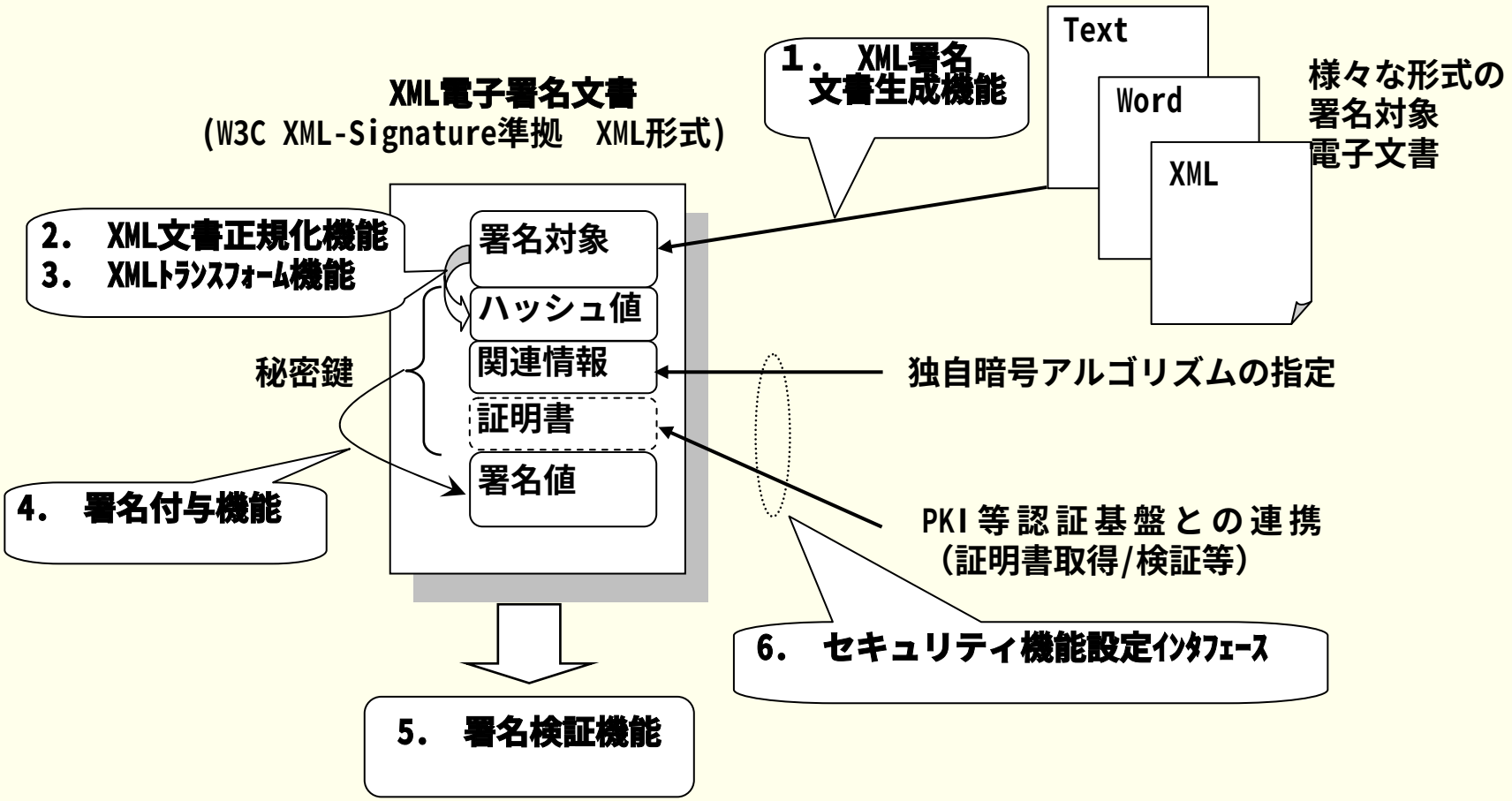
W3C XML署名に基づく署名付与と検証プロセス



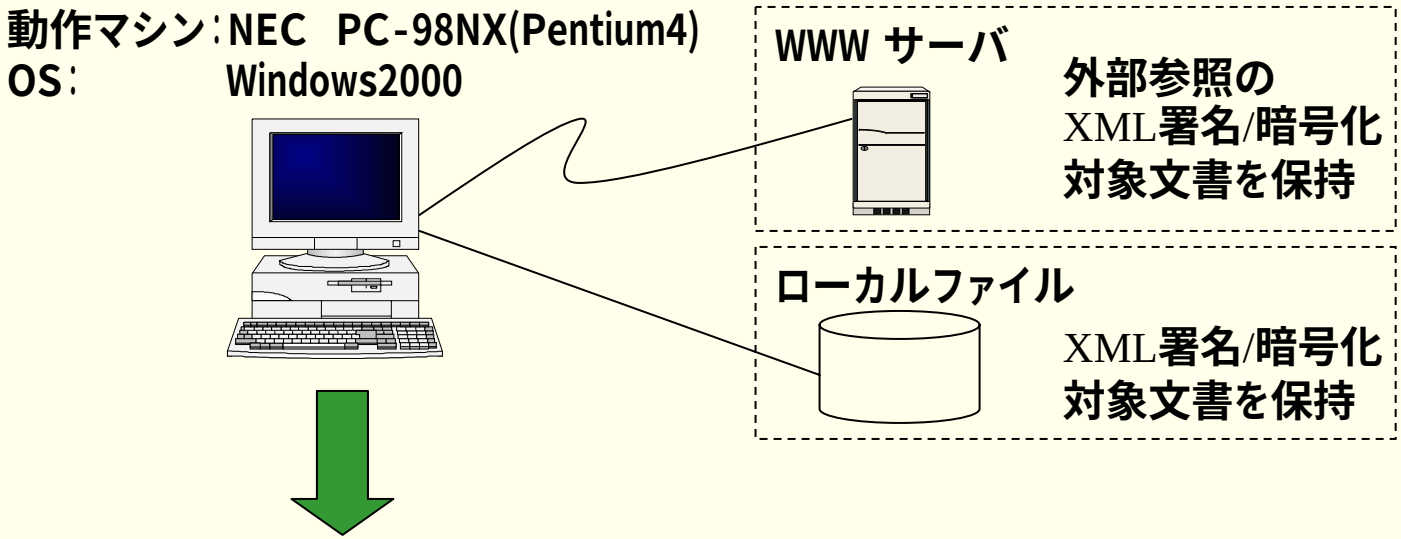
ＸＭＬ署名基盤ソフトウェアの機能一覧

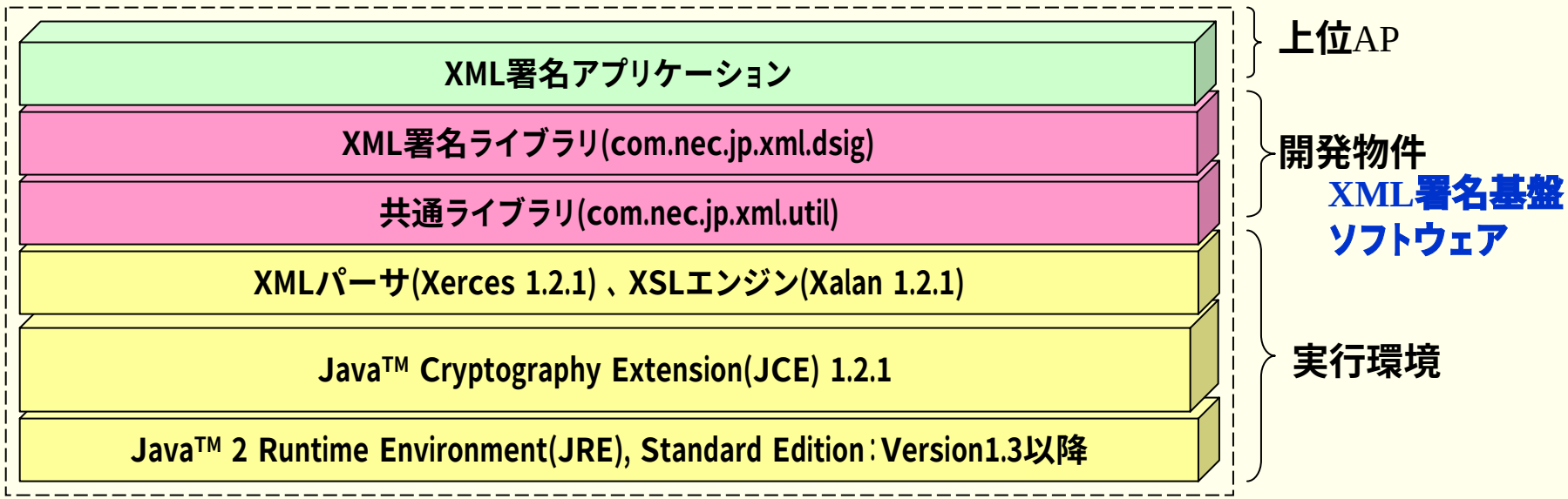
1. **XML署名文書生成機能**
   署名対象文書を指定して、XML署名文書の内部データ構造を生成する機能。
2. **XML文書正規化機能**
   XML文書に署名計算を行う際に、表現形式は異なるが意味的に同一なものとして扱えるように正規化する機能
3. **XMLトランスフォーム機能**
   署名対象文書に対して署名計算を行う前に、ユーザが指定するXML文書変換機能。
   Base64ｴﾝｺｰﾄﾞ・ﾃﾞｺｰﾄﾞ、XML部分抽出、XML変形アルゴリズム等が指定可能。
4. **署名付与機能**
   XML署名文書に対して署名付与する機能。正規化機能、ﾄﾗﾝｽﾌｫｰﾑ機能を経て得られた電子文書のハッシュ値を求める。ハッシュ値を秘密鍵で暗号化することにより署名値を求める。
5. **署名検証機能**
   XML署名文書に対して署名値を検証する機能。署名値を署名者の公開鍵で複合化、4のハッシュ値と一致するかを検証する。
6. **セキュリティ機能設定インタフェース**
   ユーザがXML署名の計算を行う際に，各種セキュリティ機能（暗号・ﾊｯｼｭｱﾙｺﾞﾘｽﾞﾑ等）指定を行う

ＸＭＬ署名フォーマットと機能関連図



XML署名基盤ソフトウェアの実行環境



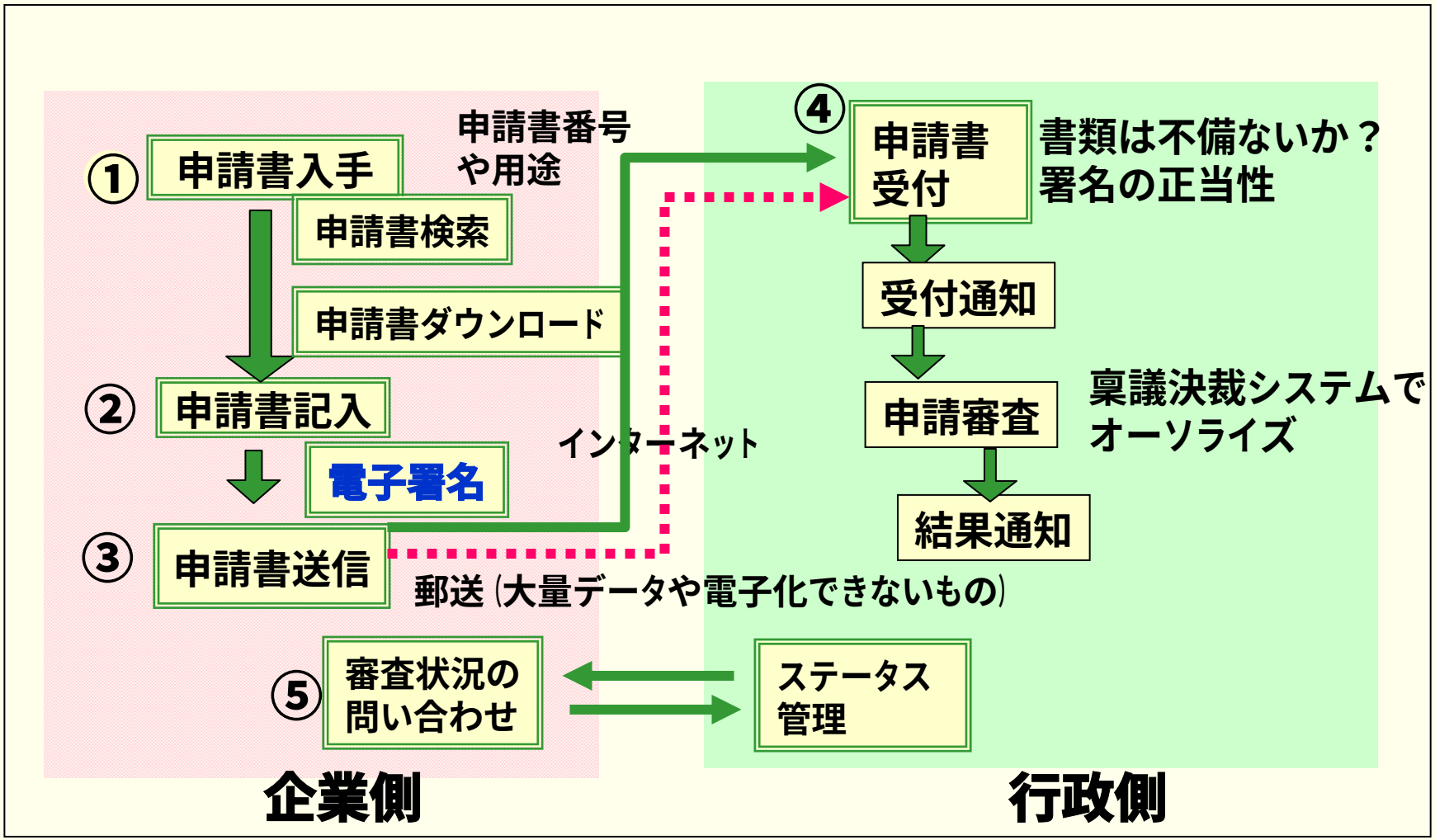


XML署名でサポートした処理アルゴリズム一覧

アルゴリズム タイプ	関連要素	アルゴリズム	W3C規約 要件	アルゴリズムURI http://www.w3.org/
ダイジェスト計算	DigestMethod要素	SHA1	REQUIRED	2000/09/xmldsig#sha1
エンコード	Transform要素	Base64	REQUIRED	2000/09/xmldsig#base64
メッセージ認証コード (共通鍵方式 署名)	SignatureMethod要素	Hmac-SHA1	REQUIRED	2000/09/xmldsig#hmac-sha1
公開鍵方式 署名	SignatureMethod要素	DSAwithSHA1(DSS)	REQUIRED	2000/09/xmldsig#dsa-sha1
		RSAwithSHA1	RECOMMENDED	2000/09/xmldsig#rsa-sha1
正規化	CanonicalizationMethod要素	正規XML(コメント付き)	RECOMMENDED	TR/2000/WD-xml-c14n-20000907#WithComments
		正規XML (コメントなし)	REQUIRED	TR/2000/WD-xml-c14n-20000907
変換	Transform要素	XSLT	OPTIONAL	TR/1999/REC-xslt-19991116
		XPath	RECOMMENDED	TR/1999/REC-xpath-19991116
		Enveloped Signature	REQUIRED	2000/09/xmldsig#enveloped-signature

III.電子署名技術の適用イメージ

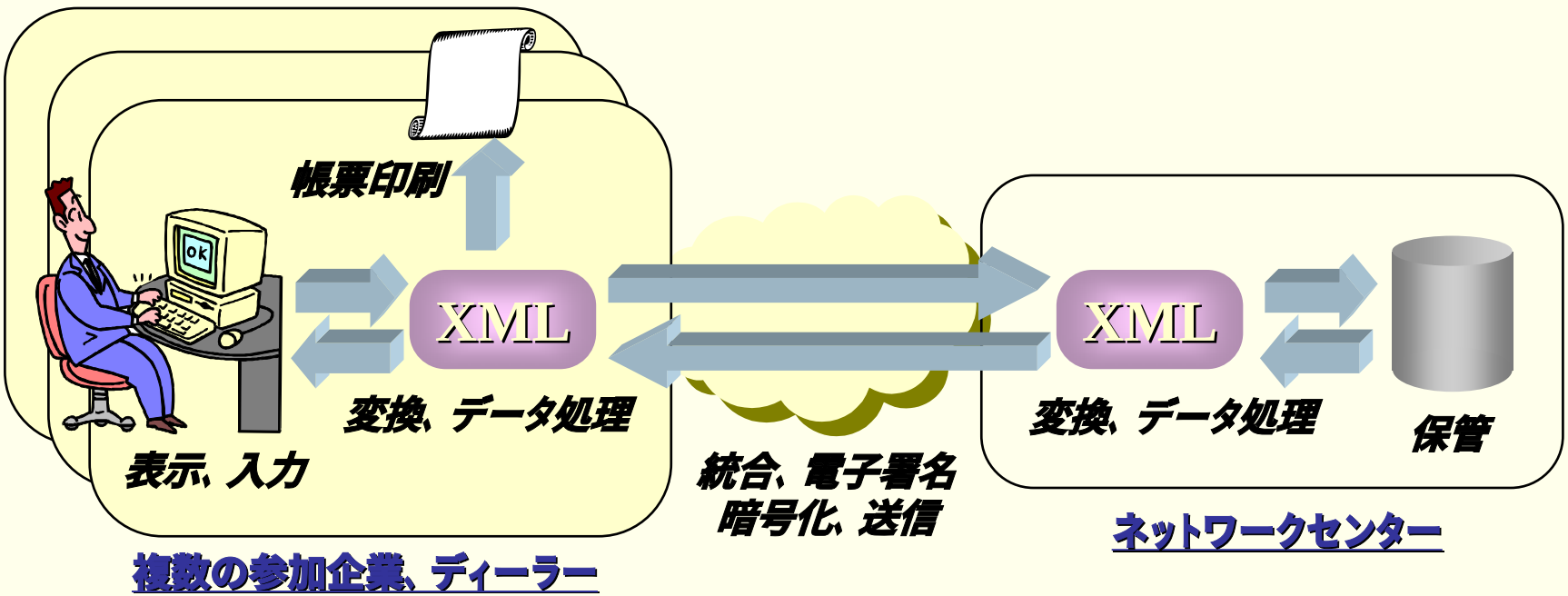
典型的な電子申請システムのフロー



XML署名のEDIへの適用例

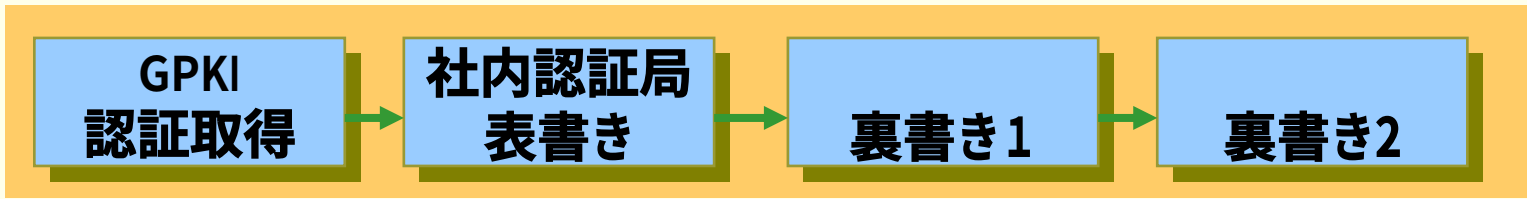
◆電子金融券面決済システム

- 電子金融券面のXML化による電子化と取引の効率化
- すべての処理場面をXML化することで一貫した拡張性のあるシステム構築
- 金融取引でのXML運用に対する安全性確保(XML電子署名の採用)

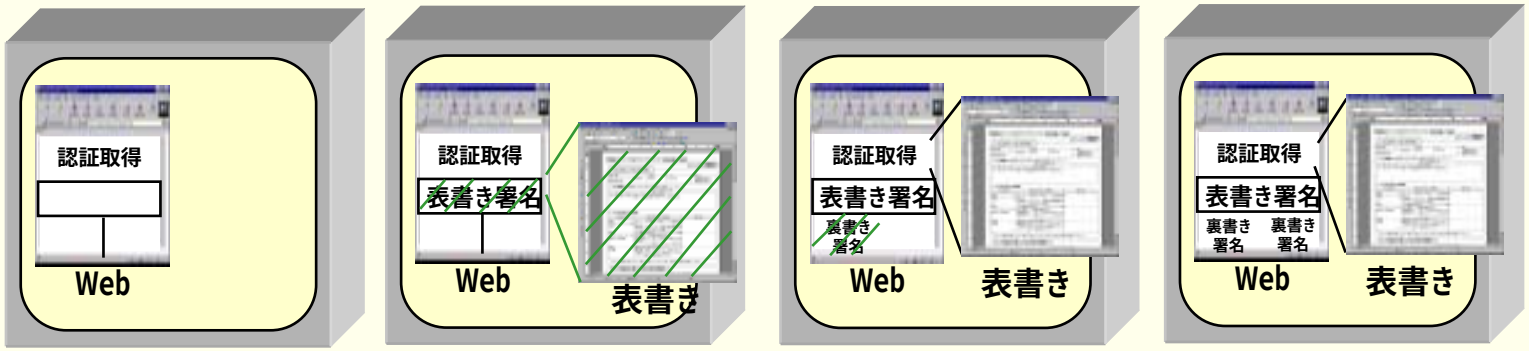


電子金融券面における多重電子署名イメージ

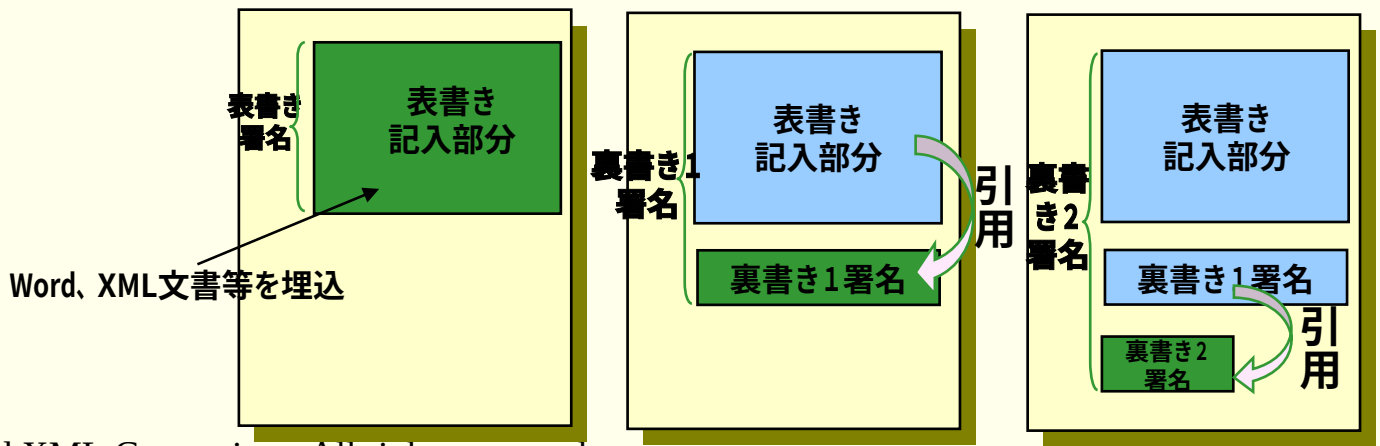
業務フロー
(フロー定義
フロー実行)



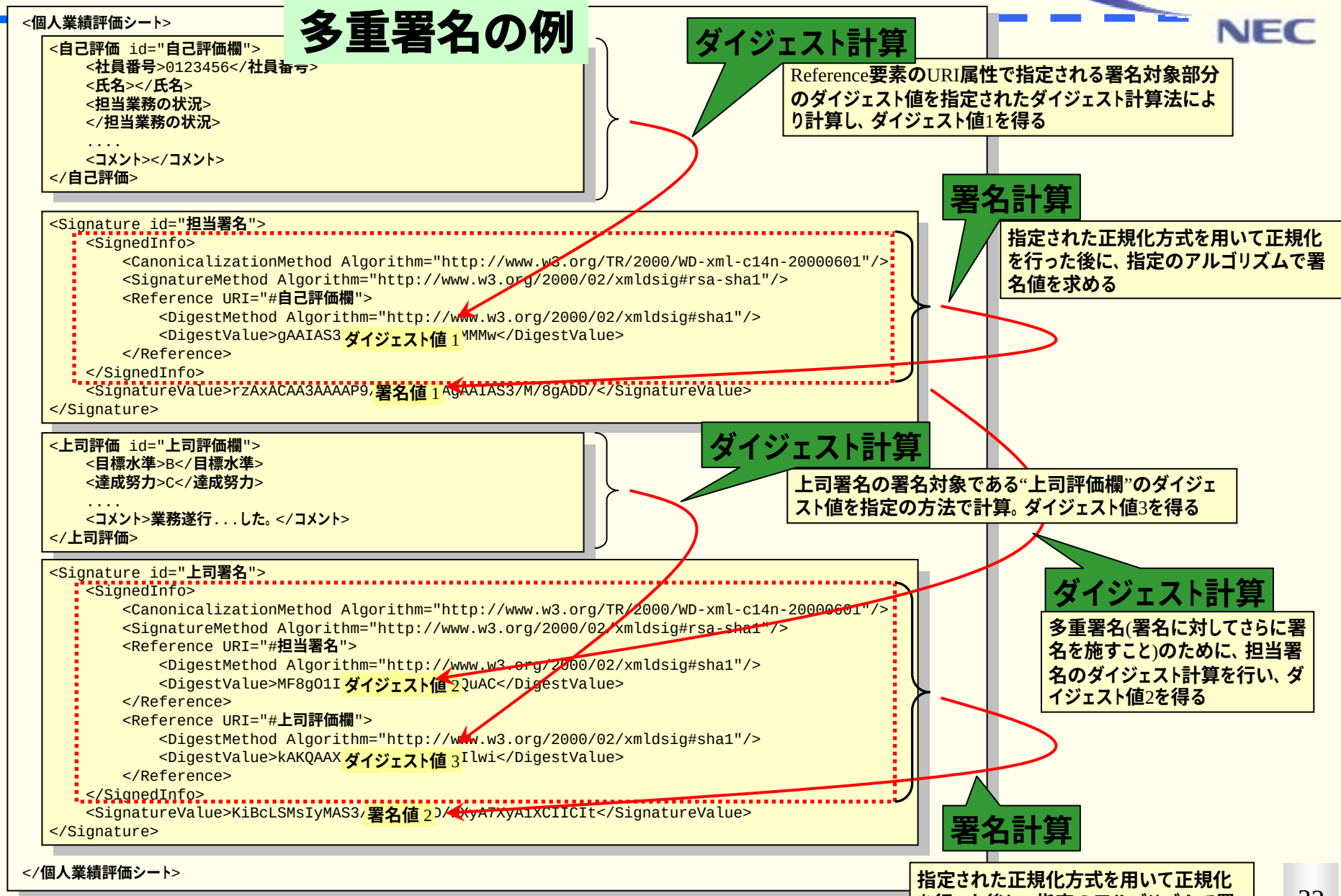
入力画面



多重署名文書モデル
(XML文書構造イメージ)



XML署名によるセキュリティ文書(多重署名) モデル



主な法定保存文書

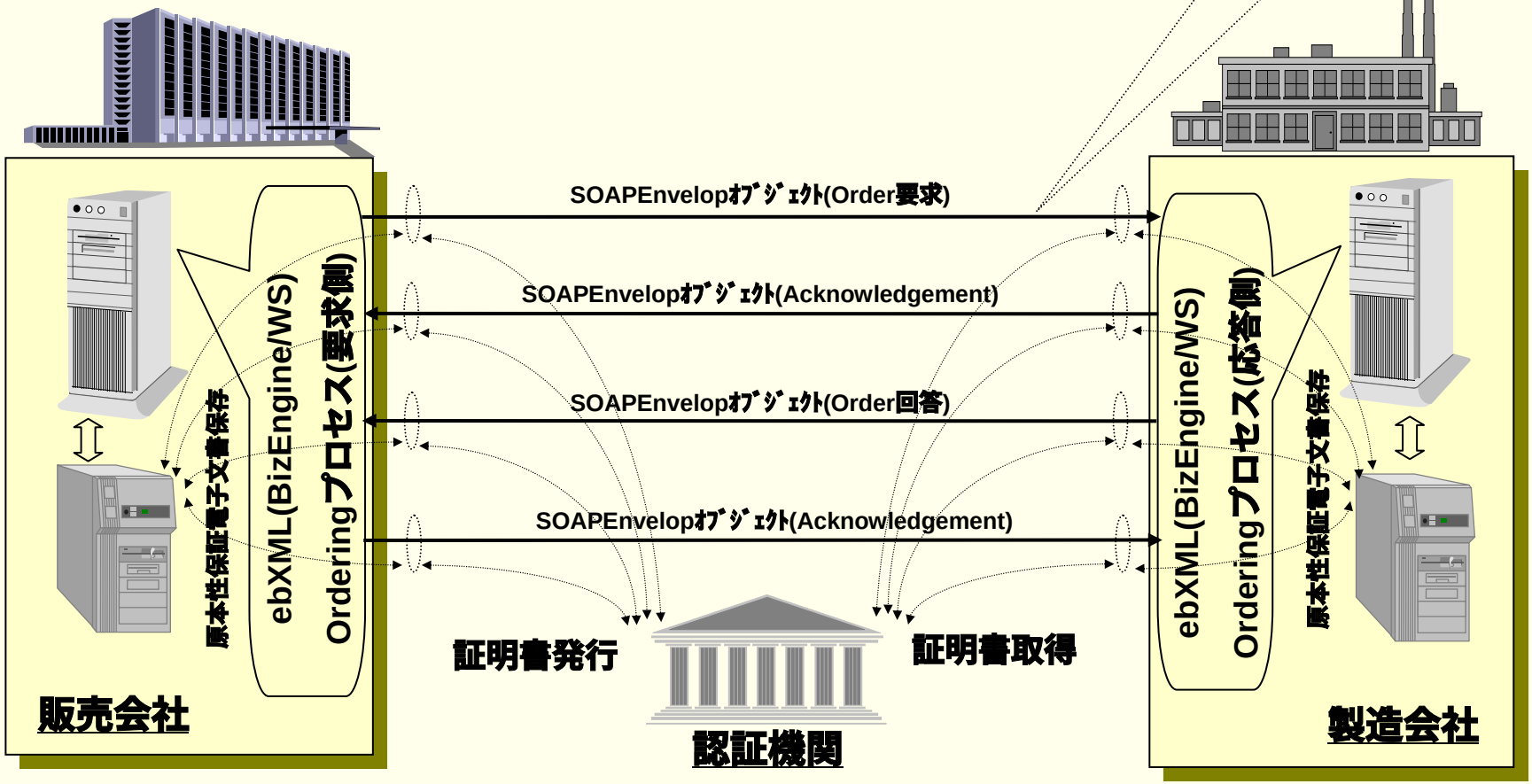
電子署名法施行に際して、以下の業務文書が電子処理可能となり、法的に指定された期間、電子文書の原本を保存しなくてはならない。

法定保存文書	保存期間	根拠となる法律・規則
商業帳簿、営業重要書類	10年	商法第36条
仕訳帳、総勘定元帳、貸借対照表,損益計算表、現金預金取引等関係書類、注文書、契約書、送り状等	７年	法人税法148、126~146、150条 所得税法施行規則第63、59~62 67条
税額控除の帳簿、請求書等	７年	消費税法第30条
診療録	５年	医師法第24条、歯科医師法23条
取締役会議事録,監査役議事録	１０年	商法第260条、商法特例法18条
雇用保険関係書類	２年	雇用保険法第143条
乗務記録（バス等）	１年	旅客自動車運送事業等運輸規則 第２５条

Webサービスセキュリティ実験編

B2B(民間業務)へのXML署名技術の拡大応用例
～ ebXML、SOAPを例として ～

1. ebXML /SOAPオブジェクト のコンテンツを通信時に完全性保証
   → XML署名技術を適用可能
   → SOAP　Body自身への影響なし
2. ebXML /SOAPｵﾌﾞｼﾞｪｸﾄの原本性を保証し、保存
   → 電子文書への公証付与に対してもXML署名を適用可能



Webサービスセキュリティ(XML署名)互換性評価

[実験環境1] クライアント
・OS：Windows2000 以降
・Java実行環境：JavaTM 2 Runtime 1.3
・ライブラリ：OCF(Open Card Framework)
・XMLパーサ：Xerces Java 1.4.0 以降
・XSLTエンジン：Xalan Java 2.1.0 以降

[実験環境2] サーバ
・OS：SolarisTM 8 (Intel Platform版) 以降
・WWWサーバ：Apache 2.0相当
・Java実行環境：JavaTM 2 Runtime 1.4
・XMLパーサ：SUN. JAXP1.1 RI
・XSLTエンジン：SUN. JAXP1.1 RI

[評価内容]

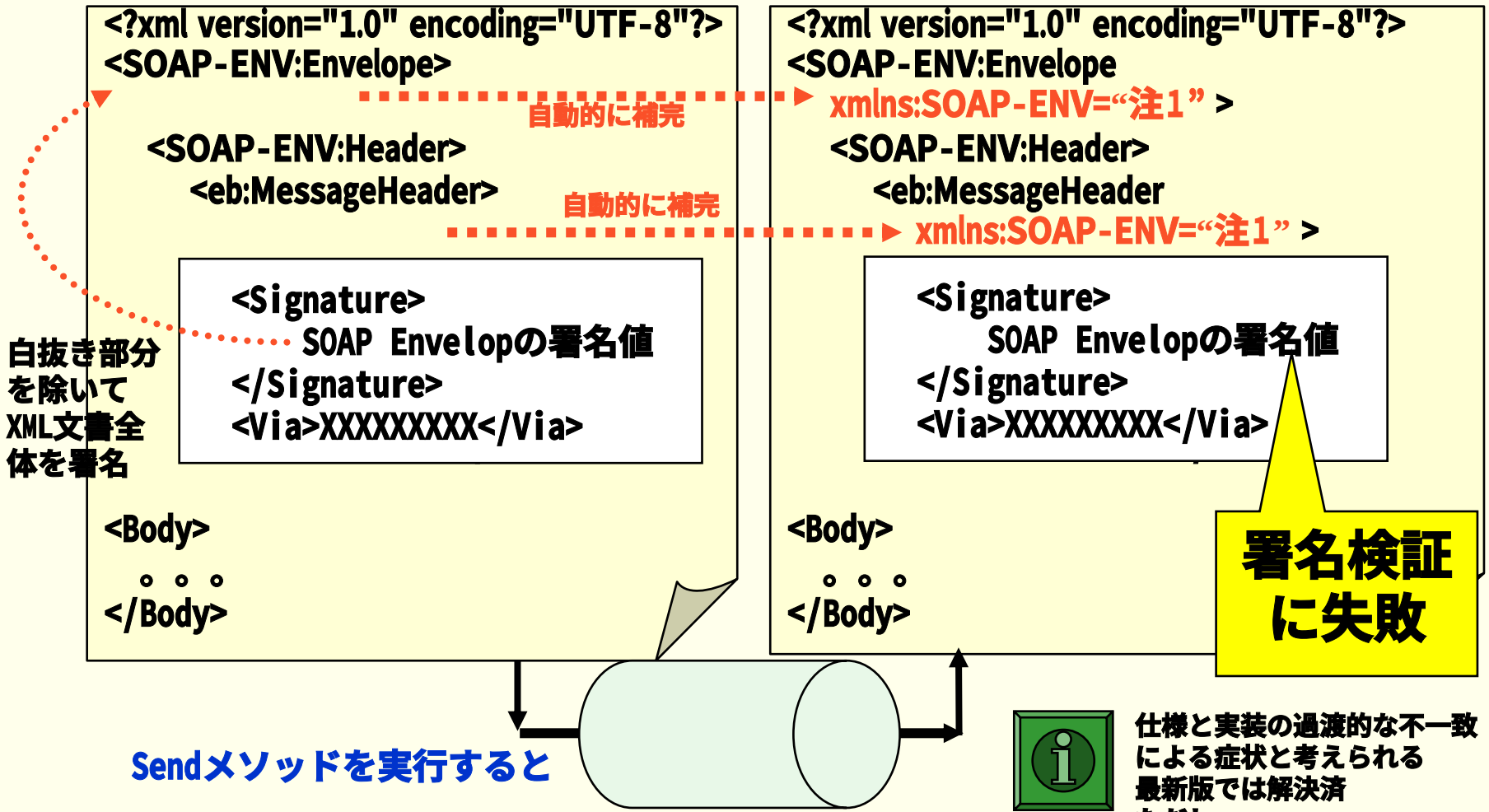
1. XML処理ツールの相互運用性の検証

・XMLパーサ、XSLT、Xpathにおける互換性検証

2. 日本語文字コードの相互運用性の検証

・異なる文字符号化スキーム (Shift_JIS、EUC-JP、UTF-8) の署名対象文書および署名検証文書中の文字コードがUnicodeに変換される際の互換性検証

Apache SOAP2.0に起因するXML署名非互換要因



注1＝ http://schemas.xmlsoap.org/soap/envelope/

実験結果に基づく非互換要因

1. **XML署名、Canonical XMLライブラリ実装バグ**
   (内容)仕様が難解、複雑であることも起因 → W3C WGへ指摘

2. **XML処理ツール(XMLパーサ,XSLT,Xpath)の実装バグ**
   SUN Java XML Pack Fall 01のXMLパーサ(8件):

   例) 属性値に文字参照を使用した際の正規化バグ、不正URL表記の未排除

   Xerces Java Parser 1.4.4(5件):

   例) 不正URL表記の未排除

3. **XML1.0仕様の実装依存部におけるXML処理ツールの実装差異**
   XMLパーサ(1件)
   例) XML文書ヘッダ部のDTD情報をDOM内部保持する(Xerces)／しない(Sun)

   注)非妥当性検証モードでの外部実体参照解決を行う(Xerces ,Sun)／行わない(ものもある)

4. **文字符号化スキーム(Shift-JIS,EUC-JP)からUnicodeへの変換表の実装差異**
   Shift_JIS→Unicode(4種類)
   ¥、~、ー、＼、〜、‖、―、¢、ポンド記号、¬、NEC特殊文字、IBM拡張文字、外字
   EUC-JP→Unicode(6種類)
   逆斜線、~、ー、＼、〜、‖、―、¥、¢、ポンド記号、¬
   (文字符号処理系が採用している変換表により上記の文字符号がUnicodeに変換された際のコード値が異なる)

XMLパーサの実装バグによって発生する非互換事例

処理命令(PI)記述の処理に関するXML1.0仕様

- [28] doctypedecl ::= '<!DOCTYPE' S Name (S ExternalID)? S? ('[' (markupdecl ¦ DeclSep)* ']' S?)? '>'
  [29] markupdecl ::= elementdecl ¦ AttlistDecl ¦ EntityDecl ¦ NotationDecl ¦ PI ¦ Comment
- 処理命令は文書の文字データの一部ではないが、アプリケーションに渡さなければならない。
  → XMLパーサは文書型宣言の内部サブセット内に記述された処理命令を処理しなくてはならない。

入力のXML文書

```
<!DOCTYPE doc [
<?Pi?><!--comment-->
<!ELEMENT doc EMPTY>
<?Pi?><!--comment-->
<!ATTLIST doc att CDATA #IMPLIED>
<?Pi?><!--comment-->
<!ENTITY % ent "">
<?Pi?><!--comment-->
<!NOTATION not PUBLIC "some notation">
<?Pi?><!--comment-->
]>
<doc/>
```

Xerces Java 1.4.4 → C14N →

```
<?Pi?>
<?Pi?>
<?Pi?>
<?Pi?>
<?Pi?>
<doc></doc>
```

文書型宣言の内部サブセット内に記述された処理命令を解釈しており、正しく動作している。

JAXP1.1 RI → C14N → <doc></doc>

文書型宣言の内部サブセット内に記述された処理命令の解釈を行っておらず、不正な動作である。

XML1.0仕様の実装依存におけるXMLパーサの実装差異によって発生する非互換事例

非妥当性検証モードにおける外部実体の処理に関するXML1.0仕様

非妥当性検証モードで動作するXMLプロセッサは、
- 実体が外部実体の場合、実体の置換テキストを取り込んでもよいが、取り込むことを義務づけられてはいない。
  → 外部実体を評価するかしないかは任意である。
- 読み込んでいないパラメタ実体への参照より後に現れた属性リスト宣言を処理してはならない。

入力のXML文書

```
<!DOCTYPE doc [
<!ELEMENT doc (#PCDATA)>
<!ENTITY % e SYSTEM "file:ext.ent">
<!ATTLIST doc a1 CDATA "v1">
%e; ←①
<!ATTLIST doc a3 CDATA "v3"> ←②
]>
<doc></doc>
```

外部実体(ext.ent)

```
<!ATTLIST doc a2 CDATA "v2" > ←③
```

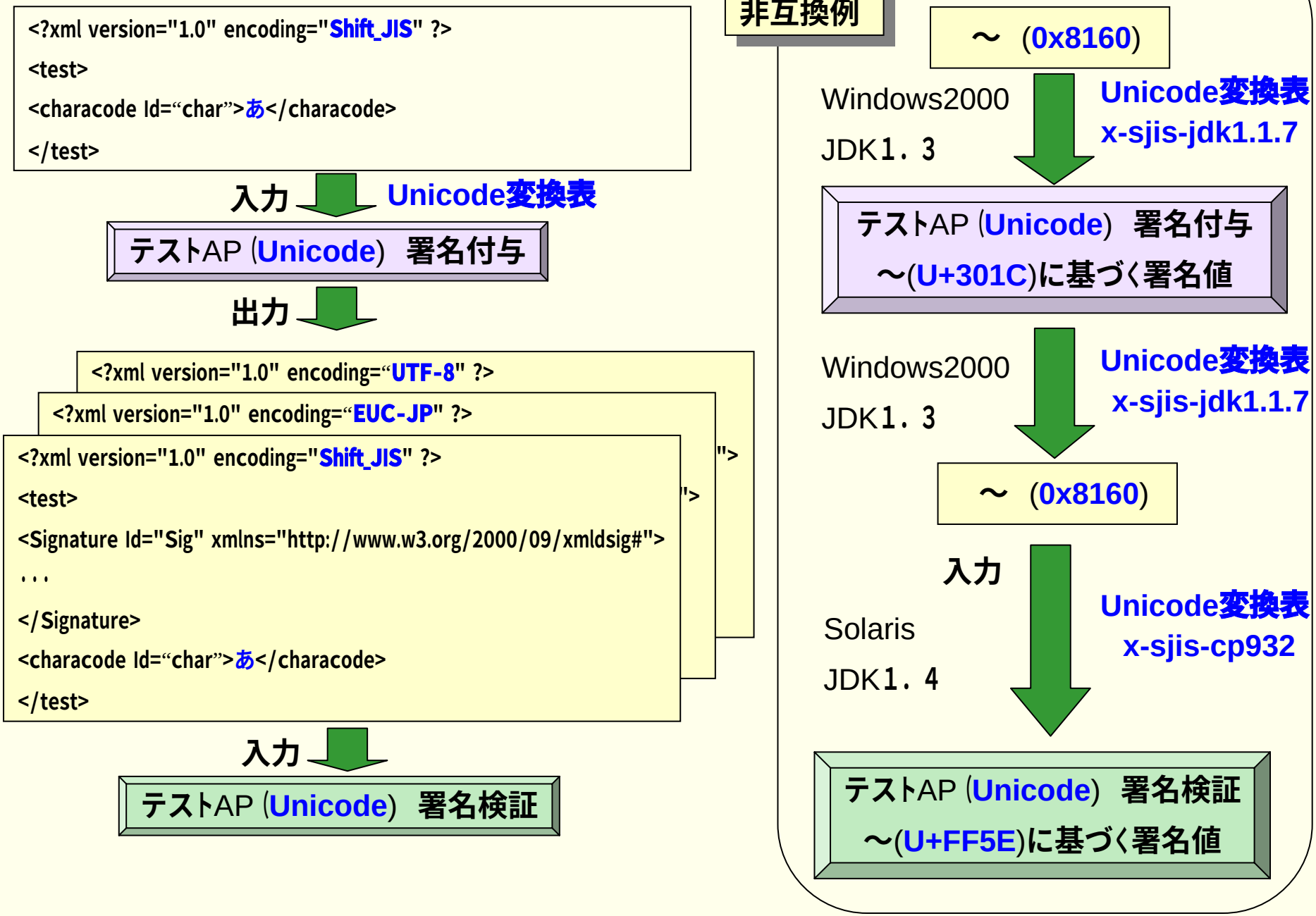
Xerces Java 1.4.4、JAXP1.1 RI → C14N → `<doc a1="v1" a2="v2" a3="v3" ></doc>`

外部実体参照①を発見した後も内部サブセットの評価を継続し、外部実体中の属性リスト宣言③、属性リスト宣言②の評価を行っている。

(XMLパーサ) → C14N → `<doc a1="v1"></doc>`

外部実体参照①を発見した時点で内部サブセットの評価を中断し、外部実体中の属性リスト宣言③、属性リスト宣言②の評価がスキップされている。

日本語文字コード非互換実験要領と非互換例



XML署名の相互運用性確保ガイドライン

[全般]

- シングルプラットフォームでシステムを構築
  XML処理ツール、Unicodeへの変換表等の統一（確実だが非現実的）

[XML処理ツール関連]

- 稼動環境が採用するXML処理ツールの事前コンフォーマンスチェックが必要
  ← 本検証実験のチェックツール・データ等が流用可能
- 2の結果に抵触しない範囲でXML文書を運用するガイドをシステム開発者にガイドライン化
- （XML署名のマルチプラットフォームを確保するためには、本格的にXML１．０完全準拠XLM処理ツールの世界的な認定機構の提言が必要）

[日本語文字関連]

- Unicode変換表をシステムで統一する、あるいは、システム内で利用するXML文書は全てUTF－8形式で保存
- １を採用不可な電子文書（既存資産、戸籍謄本用異形字等）に対しては、Base64エンコーディングすることによりXML署名を付与

今後の方針

１．XML署名・XML暗号API仕様の標準化

XML署名・暗号標準APIを標準化団体（Java Community Process等）へ提案

２．電子政府関連プロジェクトと連携

電子申請XML署名化の推進、GPKI等の認証サービス業務へ成果の紹介

３．電子商取引システムの公証機能へ適用

ebXML、SOAP等のB2Bプロトコルにおける電子署名を用いた公証機能に、XML署名を適用

４．XML署名ライブラリの製品化

W3C/IETF正式勧告準拠（2002年2月版）のXML署名ライブラリの製品化

本報告は、通信・放送機構殿研究テーマ「暗号アプリケーションプログラムインターフェース基盤技術の研究開発」 **の成果を利用しております**