セキュリティ関連XML規格の紹介

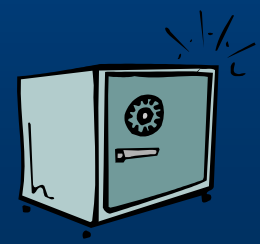
ビジネスケースを想定して. . .

2002年6月10日

XMLコンソーシアム 基盤技術部会
共通基盤WG セキュリティSWG

本日の報告内容

- セキュリティ関連XML規格をケーススタディを通して報告します
  - セキュリティとは
  - 現状適用可能な技術
  - XMLセキュリティ技術の特徴
  - ケーススタディ「旅行」
  - 図解XML規格の概略説明と XMLセキュリティの課題



メンバ紹介

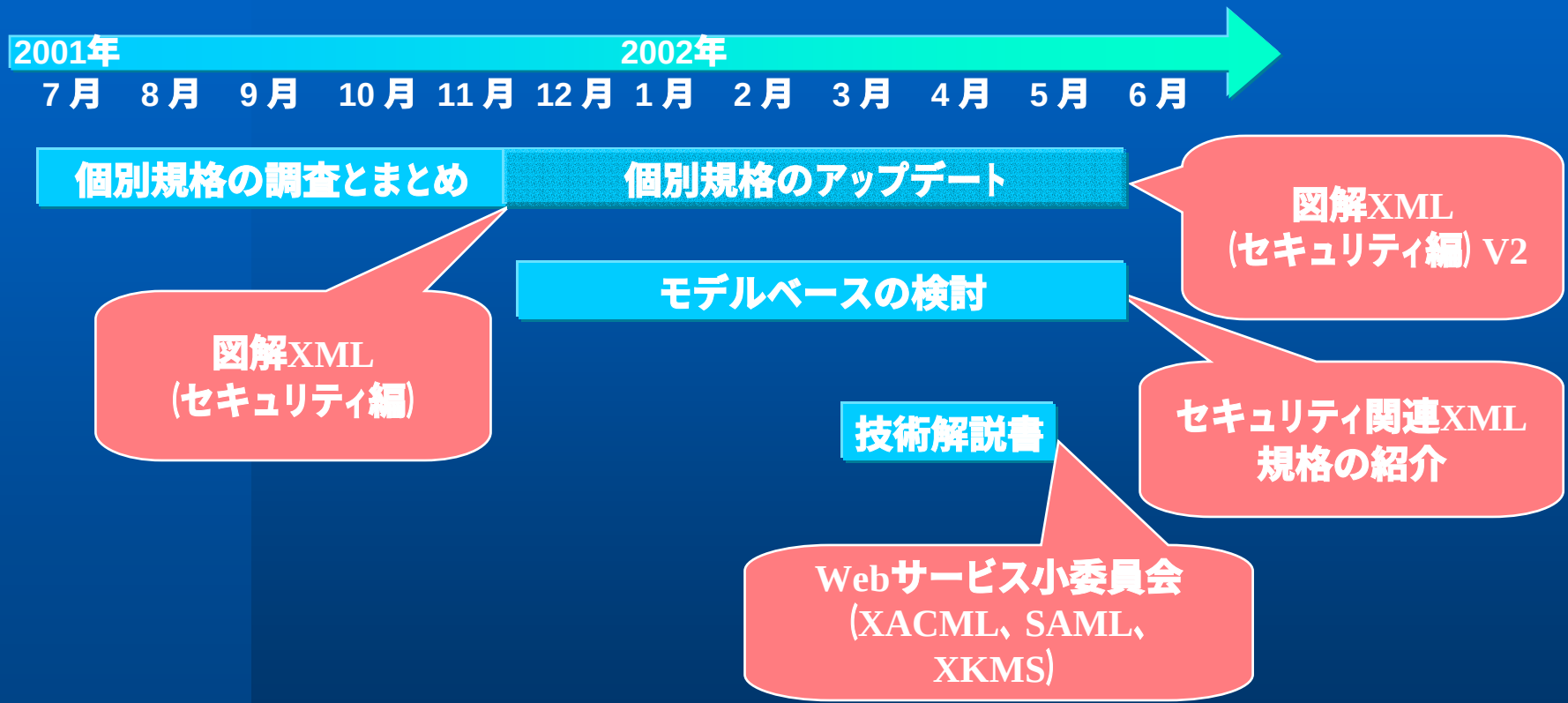
- 富士ゼロックス (株)　　道村 唯夫
- ミノルタ (株)　　上田 隆司
- 沖電気工業 (株)　　池上 勝美

背景と目標

- 背景
  - 今までは規格そのものに焦点をあてて説明
    - セキュリティの啓蒙という観点では充分な成果
  - セキュリティへの関心の高まりから質問が増加
    - 実システムでの技術適用がイメージできていない
- 目標
  - よりわかりやすくセキュリティ関連XML規格を説明することで、より広範囲の方々にわかっていただく

    → モデル化されたビジネスケースを通じて、目的志向で技術を説明

活動実績


2001年
2002年
7月 8月 9月 10月 11月 12月 1月 2月 3月 4月 5月 6月
個別規格の調査とまとめ
個別規格のアップデート
図解XML
(セキュリティ編) V2
モデルベースの検討
図解XML
(セキュリティ編)
セキュリティ関連XML
規格の紹介
技術解説書
Webサービス小委員会
(XACML、SAML、
XKMS)

セキュリティとは

- セキュリティとは、
  サービスの提供、サービスの利用が
  安全に行えること

  この「安全」を脅かすものを「脅威」と呼ぶ

  この「脅威」を排除するためのものが
  「セキュリティ」であり、
  「技術」と「運用」の両面の考慮が必要

セキュリティとは (続き)

- 脅威と対策

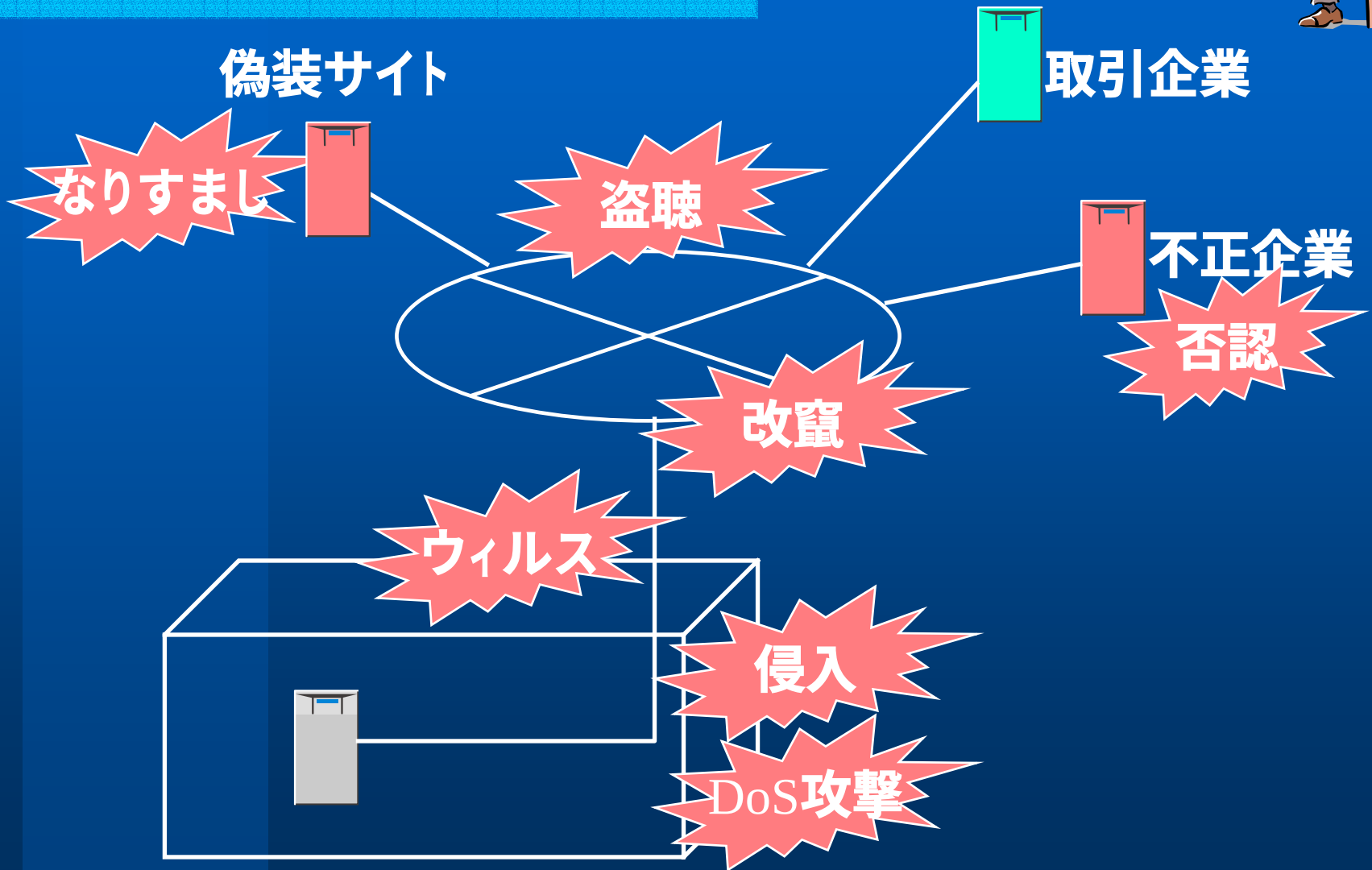
技術

- 攻撃: 提供者・利用者による破壊行為による
  - 物理的アクセスの制限と許可
  - 信頼関係の構築 (認証、与信、権限の委譲と伝播、格付け)
  - 可視・不可視データの埋め込みと検出
  - 盗聴、改竄、なりすましの防止
  - 監視、検知、監査 (障害、侵入、ウィルス、破壊)とリカバリ

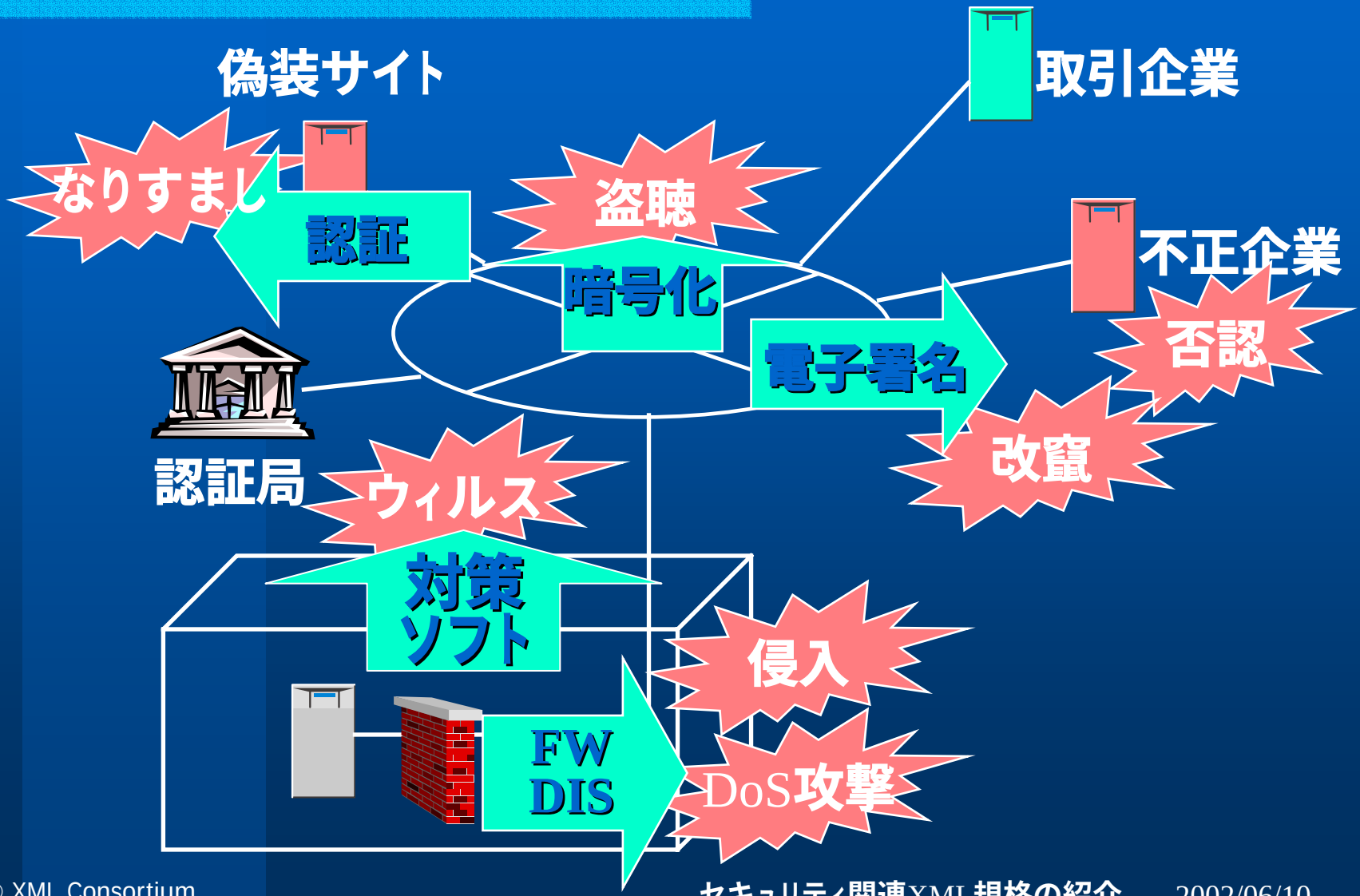
運用

- 事故: 本来想定してない事象の発生による
  - 正しい運用によって回避されることがほとんど
  - システム提供者は、「管理・運用の手助けとなる機能 (マニュアルの整備、監査機能装備など)の提供」と「強固なシステムの構築(プログラム/プラットフォームの事故の防止)」に留意する必要がある

インターネットの脅威



利用可能な技術


偽装サイト
取引企業
なりすまし
認証
盗聴
暗号化
不正企業
否認
電子署名
改竄
認証局
ウィルス
対策
ソフト
侵入
FW
DIS
DoS攻撃

XMLセキュリティ技術の特徴

- インターネット＋**相互接続**に焦点を当てた技術
  - W3C、OASIS等で規格化
- 従来の規格を**より便利**に
  - 例えば．．．．．
    - SAML→Single Sign Onを実現する
    - XACML→リソース権限を明確化する
    - XML Signature/Encryption→部分の概念を導入する
    - XKMS→クライアントの処理を軽減する
    - XML Pay→支払い処理を標準化する

ケーススタディで．．．

ケーススタディ

下田さんの旅行．．．．．

ケーススタディ「旅行」

- あらすじ
  - オリンピックカメラに勤務する東京都在住の下田さんが、インターネットを通じて企画、予約した旅行を、クレジットカードで決済する。

ステップ

1. 旅行代理店が旅行を**企画**
2. 下田さんが旅行を**予約**
3. 下田さんが予約を確認
4. 下田さんがカード会社に**支払**
5. 下田さんが友人と旅行

主役の下田さんとお友達

ケーススタディ「旅行」(続き)

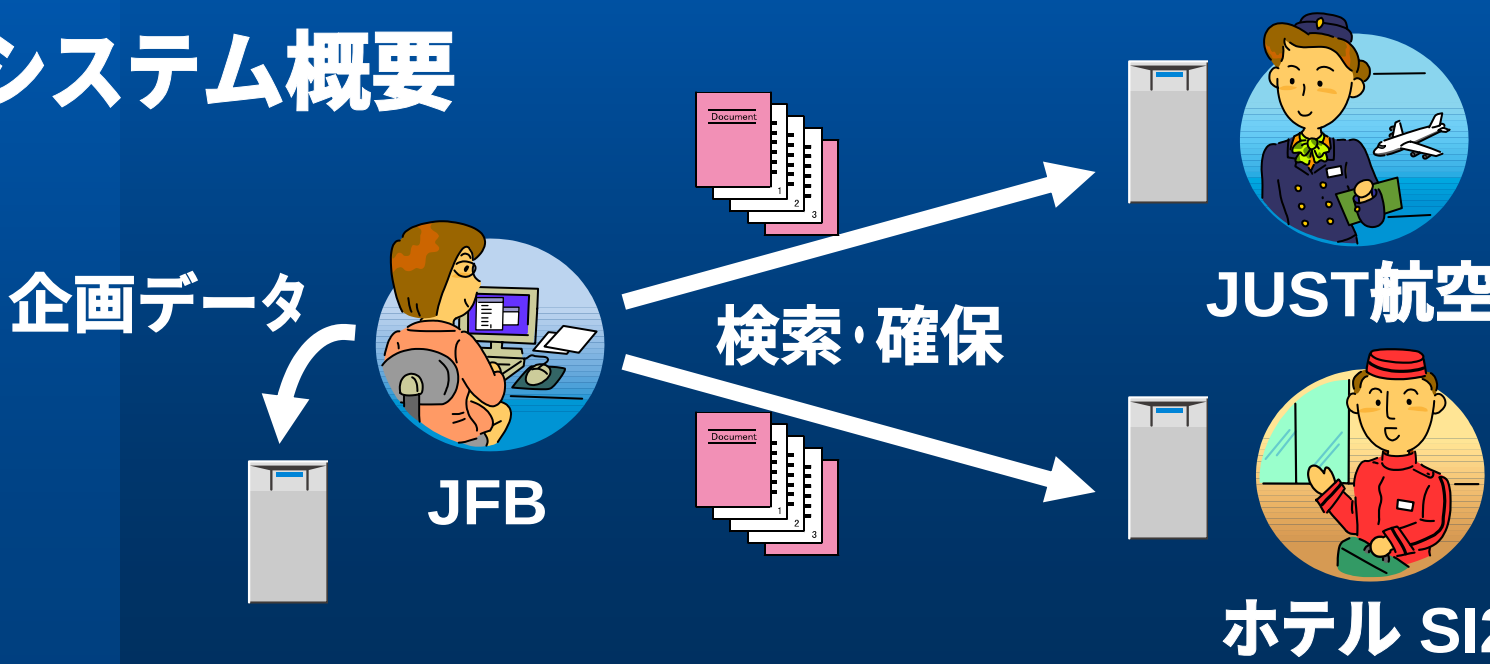
- 登場人物/団体
  - 旅行者: 下田さん [オリンピックカメラ勤務]と友人
  - 代理店: JFB
    [yasuitabi.comという企画旅行予約サイトを運営する大手旅行会社]
  - 航空会社: JUST航空
  - ホテル: ホテル SI2
  - カード会社: BEGINNERカード社

ケーススタディ「旅行:企画」

1. 旅行代理店が旅行を企画

   JFBは、JUST航空やホテルSI2のWebサイトなどを参照して企画をたてる。JUST航空などのサイトには、空き情報(条件、料金など) がある。

- システム概要



ケーススタディ「旅行:企画」

- システム要件
  - JFB
    - 複数のサイトの参照時いちいちログオンし直したくない
      → SAML
  - JUST航空、ホテルSI2
    - 大口顧客には優先的に特別な条件で提供するが、一般の顧客にはその条件を知られたくない
      → XACML, XML Encryption
    - 予約の受付データは信頼のおけるものでないとダメ
      → XML Signature

SAML - Security Assertion Markup Language

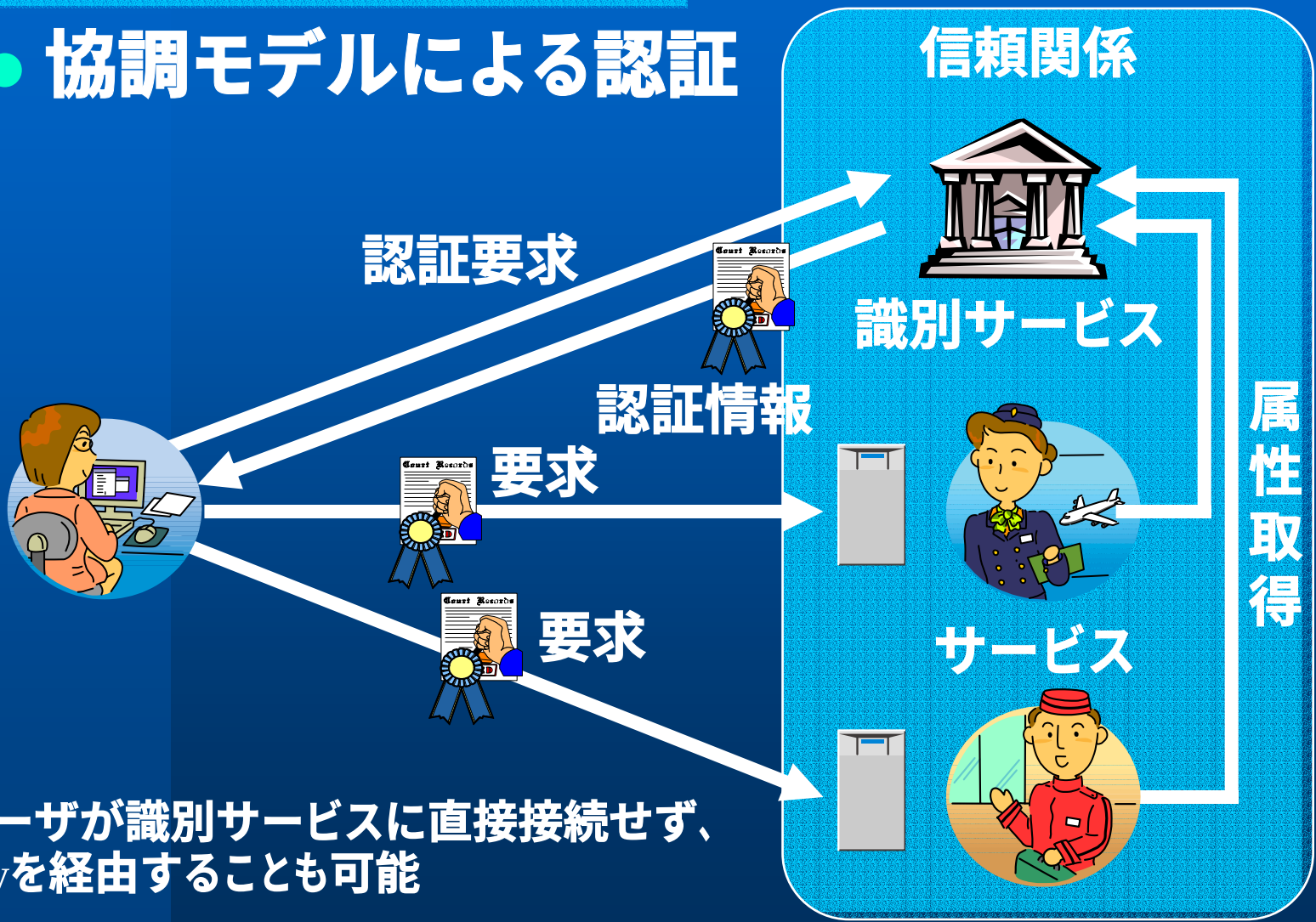
- 認証/承認/属性情報を交換するための言語
  - 最初の一回の認証だけで、複数のサーバ資源が使用できる(Single Sign-On)ようにできる
    - アサーションの表現方法とプロトコルの定義
    - 広範な既存認証機構との連携
    - 三つのモデル(Pull/Push/3rd Party Security)を想定
  - OASIS (e-business分野における標準化推進のための業界団体[NPO]) において制定中
  - 2002年7月以降にOASIS標準になる予定

詳細については...

OASIS SSTC: http://www.oasis-open.org/committees/security/

SAML（続き）

- 協調モデルによる認証



※ユーザが識別サービスに直接接続せず、Proxyを経由することも可能

SAML（続き）

- 認証要求の例

```
<samlp:Request MajorVersion="1" MinorVersion="0“
    RequestID=“8xtyzzKqPMLcFswefRIJAL">
    <samlp:RespondWith>AuthenticationStatement</samlp:RespondWith>
    <samlp:AuthenticationQuery>
      <saml:Subject>
        <saml:NameIdentifier Name=“agent@JFB.co.jp"/>
        <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（続き）

- 認証応答の例

```
<samlp:Response InResponseTo="8xtyzzKqPMLcFswefRIJAL"
    MajorVersion="1" MinorVersion="0"
    ResponseID="xmlconsortium2002061002090011">
    <samlp:Status>
      <samlp:StatusCode Value="samlp:Success"/>
    </samlp:Status>
    <saml:Assertion AssertionID="qJcZsDTnJBPPe/OLMt"
        IssueInstant="2002-06-10T11:22:33.456" Issuer="JUSTAir"
        MajorVersion="1" MinorVersion="0">
      <saml:Conditions
          NotBefore=" 2002-06-10T11:22:33.466"
          NotOnOrAfter=" 2002-06-10T15:22:33.466 "/>
      <saml:AuthenticationStatement
          AuthenticationInstant=" 2002-06-10T11:22:33.106"
          AuthenticationMethod="http://www.oasis-open.org/.../password">
        <saml:Subject>
          <saml:NameIdentifier Name="agent@JFB.co.jp"
    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
ユーザ識別子

XACML - eXtensible Access Control Markup Language

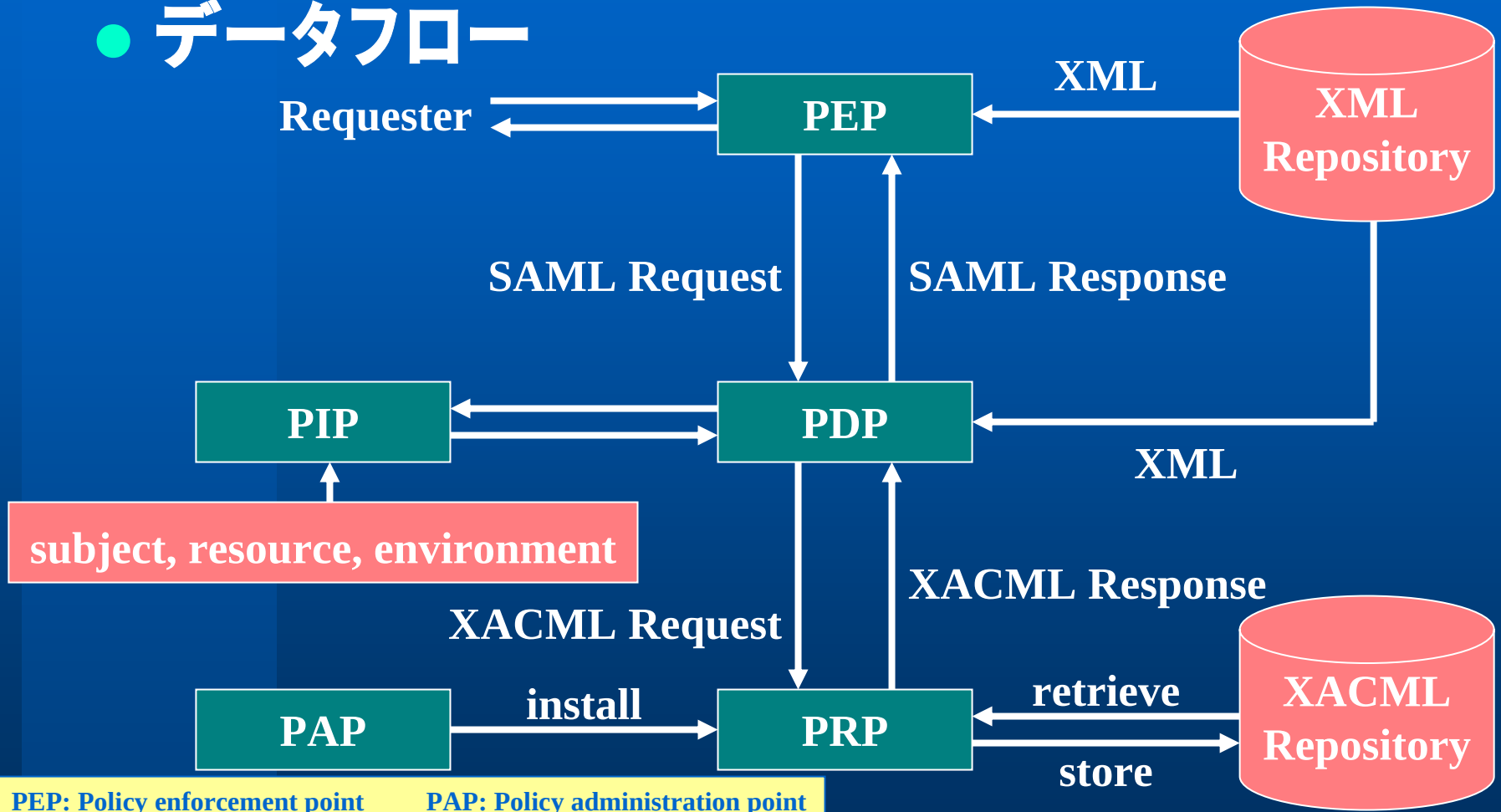
- アクセス制御ポリシーを表現するための言語
  - 様々な資源へのアクセス要求が許可されるか否認されるかといったルールを記述することが可能
  - OASIS(e-business分野における標準化推進のための業界団体[NPO]) において制定中
  - 2002年7月以降にOASIS標準になる予定

詳細については...

OASIS XACML TC: http://www.oasis-open.org/committees/xacml/

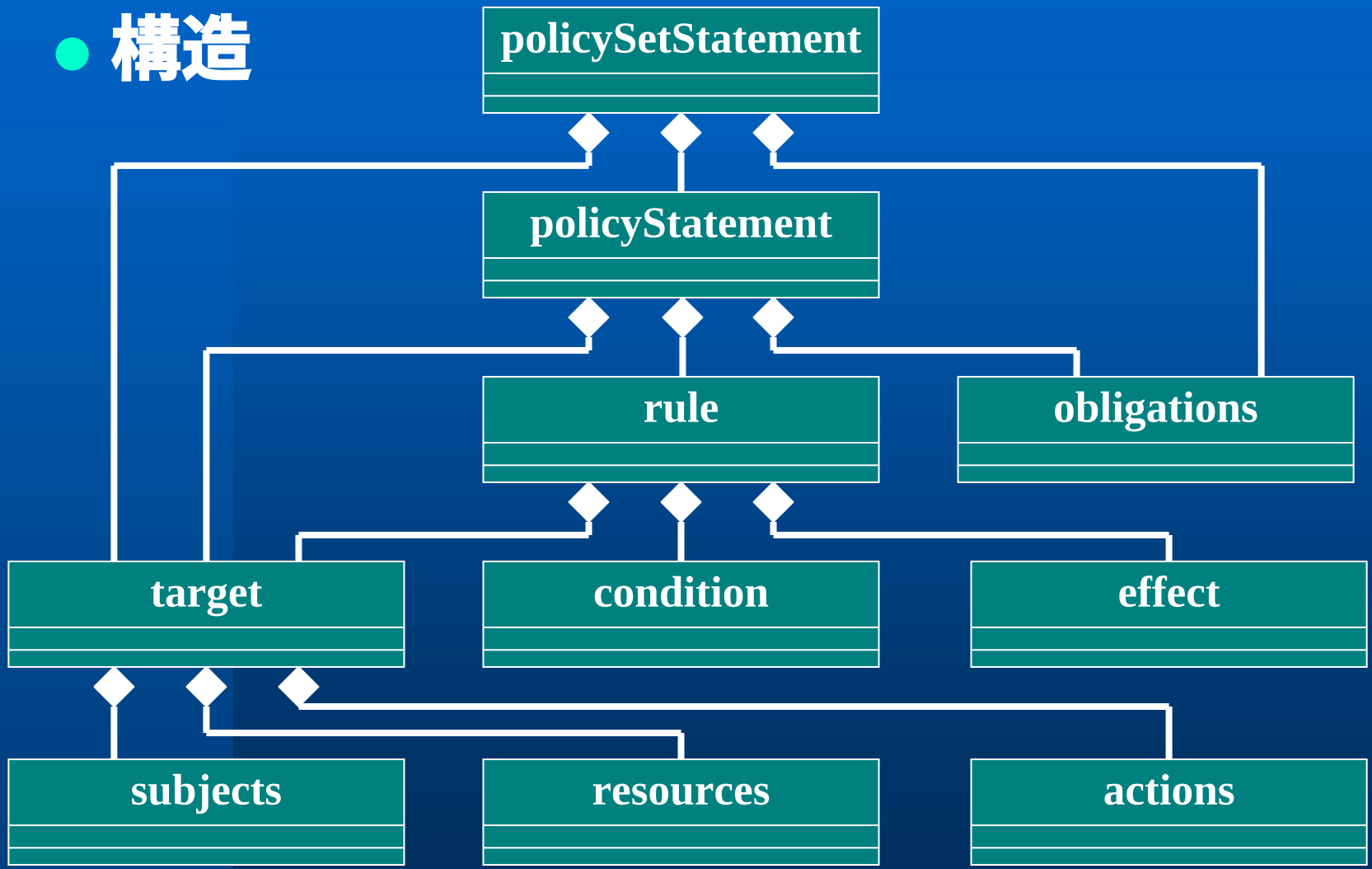
XACML（続き）

- データフロー



XACML（続き）

- 構造


policySetStatement
policyStatement
rule
obligations
target
condition
effect
subjects
resources
actions

XACML（続き）

- 構文例

```
<rule ruleId="//justair.co.jp/rule/id/1" effect="Allow">
    <description>Sample policy</description>
    <target>
      <subjects>
        <saml:Attribute AttributeName="RFC822Name">
          <saml:AttributeValue>* </saml:AttributeValue>
        </saml:Attribute>
      </subjects>
      <resources>
        <saml:Attribute AttributeName="documentURI">
          <saml:AttributeValue>//justair.co.jp/reserve/rcd.*</saml:AttributeValue>
        </saml:Attribute>
      </resouces>
      <actions>
        <saml:Action>read</saml:Action>
      </actions>
    </target>
    <condition>
      <equal>
        <saml:AttributeDesignator AttributeName="requestor"
          AttributeNamespace="//oasis-open.org/…/xacml/docs/identifiers/" />
        <saml: AttributeDesignator AttributeName="agentName"
          AttributeNamespace="//justair.co.jp/record/agent/" / >
      </equal>
    </condition>
</rule>
```

subjects
対象となる要求主体

resources
対象となるオブジェクト

actions
許可される動作

condition
許可される条件

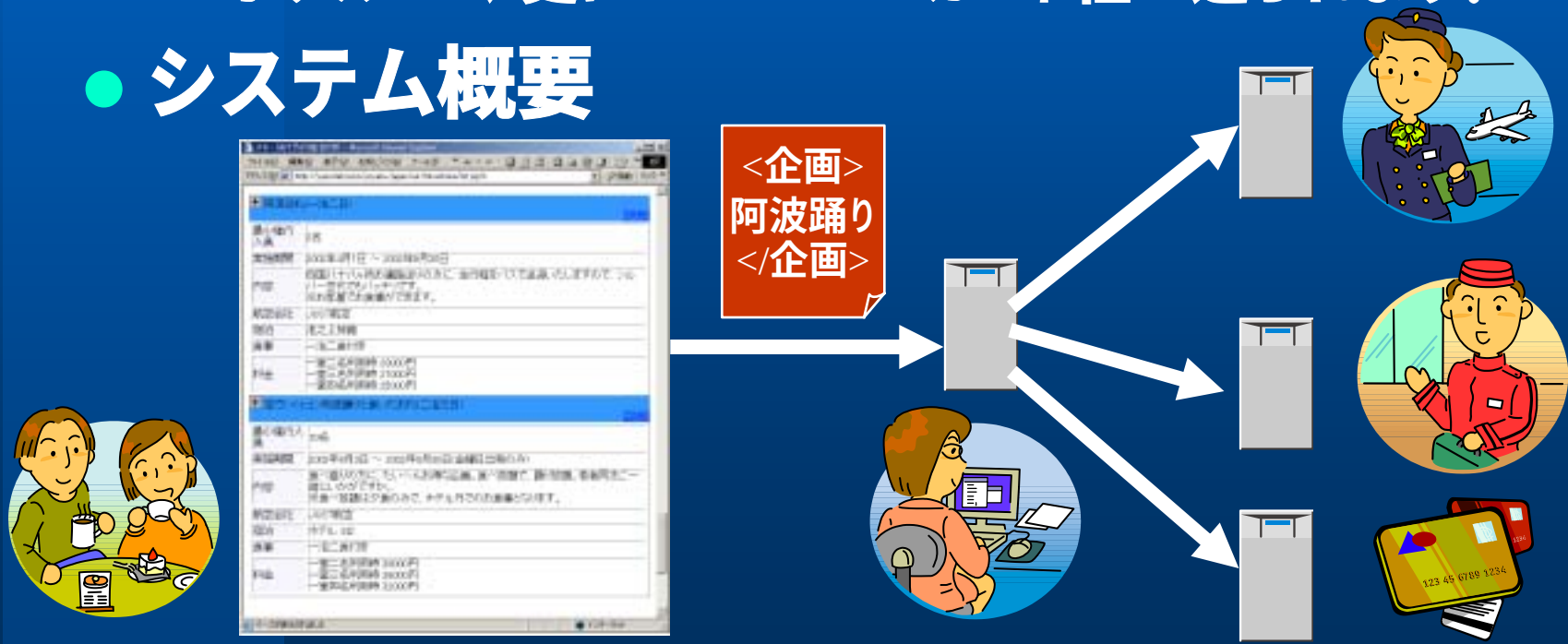
この例の場合、
「要求者が代理店の名前を持つ」

ケーススタディ「旅行:予約」

2. 下田さんが旅行を予約

下田さんはWeb画面から企画「空でいくXコン阿波踊りと食いだおれ」を予約します。入力情報はJFBからJUST航空、ホテルSI2、更にBIGINNERカード社へ送られます。

- システム概要



ケーススタディ「旅行:予約」

- システム要件
  - JFB、JUST航空、ホテルSI2、BIGINNERカード
    - 予約の受付データは信頼のおけるものでないとダメ
      - 改ざんが無いこと
      - 送信者を特定できること

      → XML Signature、XKMS
  - 下田さん
    - クレジットカードの番号はJFB、JUST航空、ホテルSI2に知られたくない
      - コンテンツの一部を暗号化できること

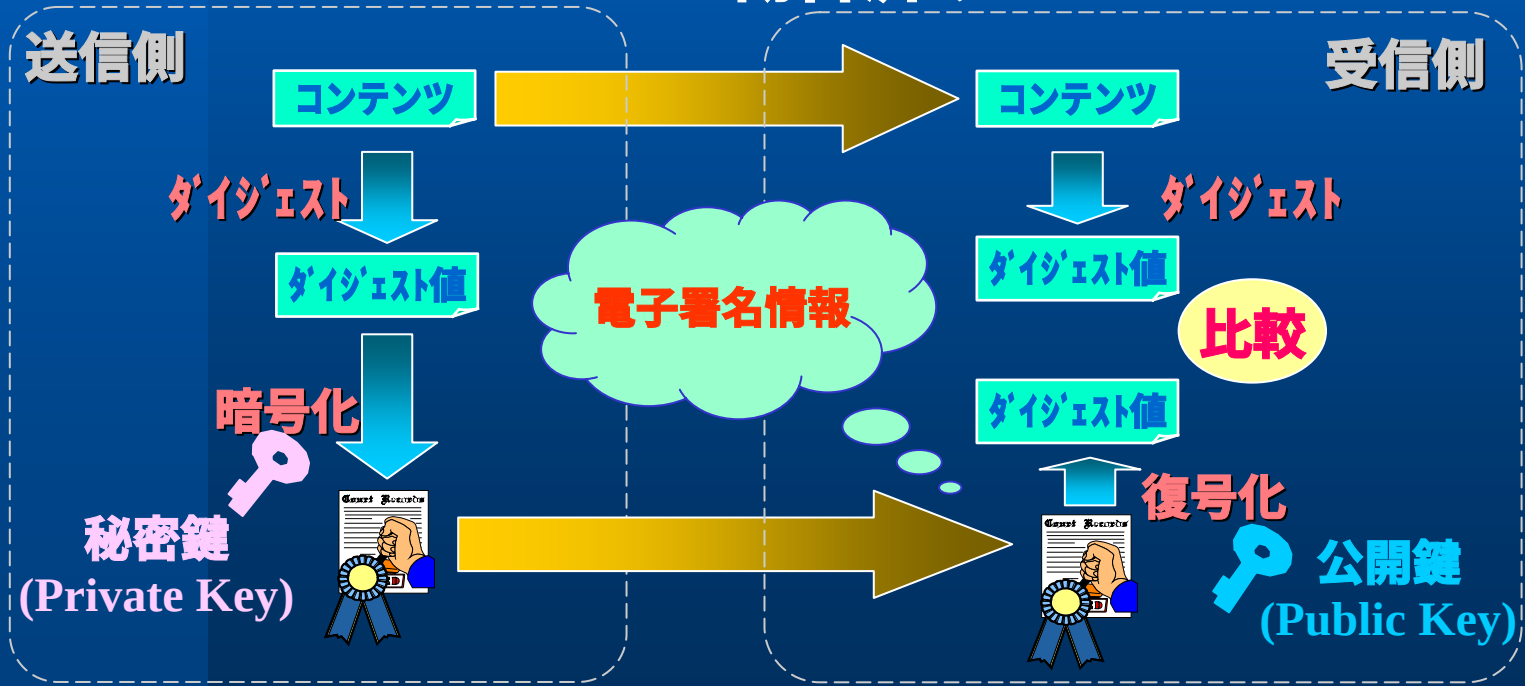
      →XML Encryption、XKMS

セキュリティの基礎技術

- ダイジェスト
  - コンテンツから一意のデータ列を生成するアルゴリズム
- 公開鍵暗号
  - 公開鍵と私有鍵の鍵ペアはユニーク
  - 一方の鍵で暗号化したデータは他方の鍵でのみ復号
    - 電子署名：送信側は私有鍵で暗号、受信側は公開鍵で復号
      →この公開鍵で復号できるデータを作成できるのは送信側のみ
    - 暗号化：送信側は受信者の公開鍵で暗号、受信者は自分の私有鍵で復号
      →特定者（受信者）のみ復号可能
- PKI
  - 公開鍵は予め認証局（CA）に登録する
    →電子署名：受信側が送信者の公開鍵を検証
    →暗号化：送信側が受信者の公開鍵を取得

XML Signature

- 電子署名に必要な情報をXMLで表現
  - コンテンツの改ざんを検出する
  - PKIと連携し、送信側の否認を防止する
  - 2002/2/12 W3Cで勧告済み



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"/>
    </CanonicalizationMethod>
    <SignatureMethod  Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-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">
    <Order><企画名>空で行くXコン阿波踊りと食いだおれ</企画名>
      <Creditcard>
        <Name>Takashi Shimoda</Name>
        <VALIDTHRU>03-05</VALIDTHRU>
        <CardNo>1234-5678-9999-0000</CardNo>
      </Creditcard>
    </Order>
  </Object>
</Signature >
```

署名対象

アルゴリズム

SignedInfo：
署名の情報

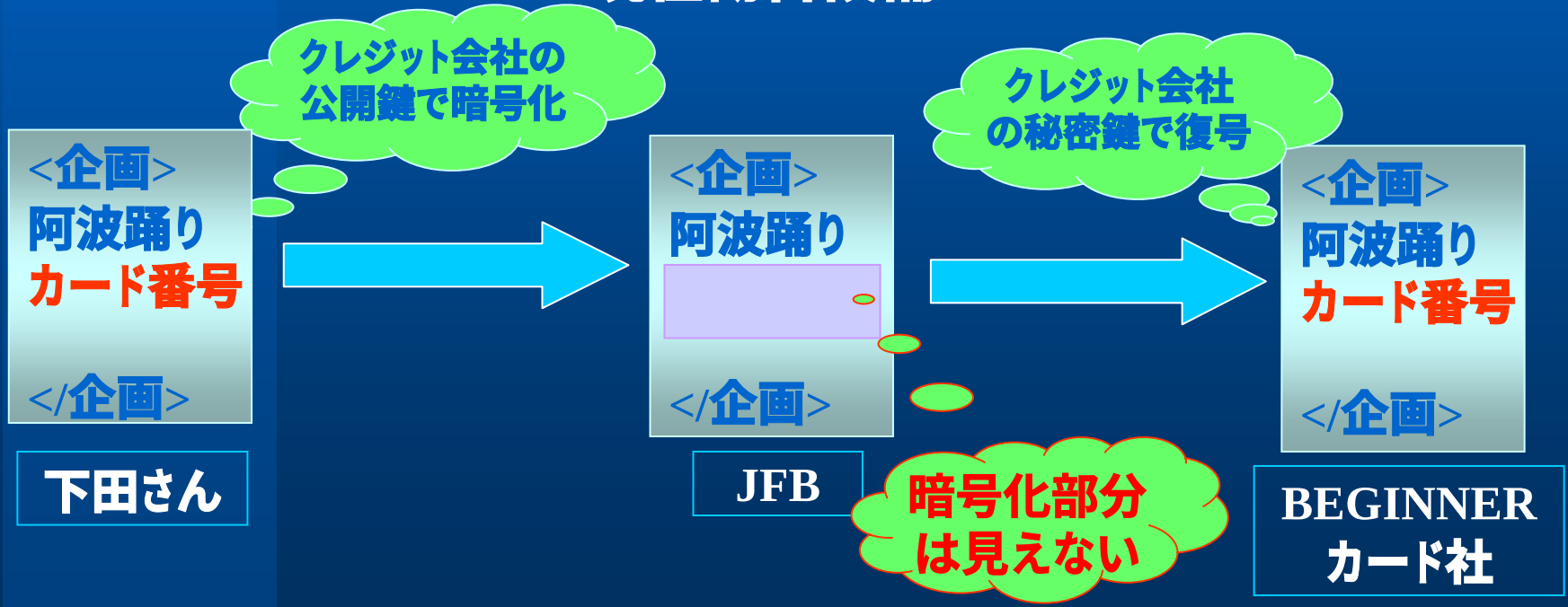
SignatureValue：
署名値

KeyInfo：
公開鍵関連情報

Object：
署名対象

XML Encryption

- 暗号･復号化の情報をXMLで表現する
  - コンテンツの一部または全部を暗号化する
  - 中継者に対するデータの部分秘匿に有効
  - 2002-03-14現在勧告候補



構文例

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

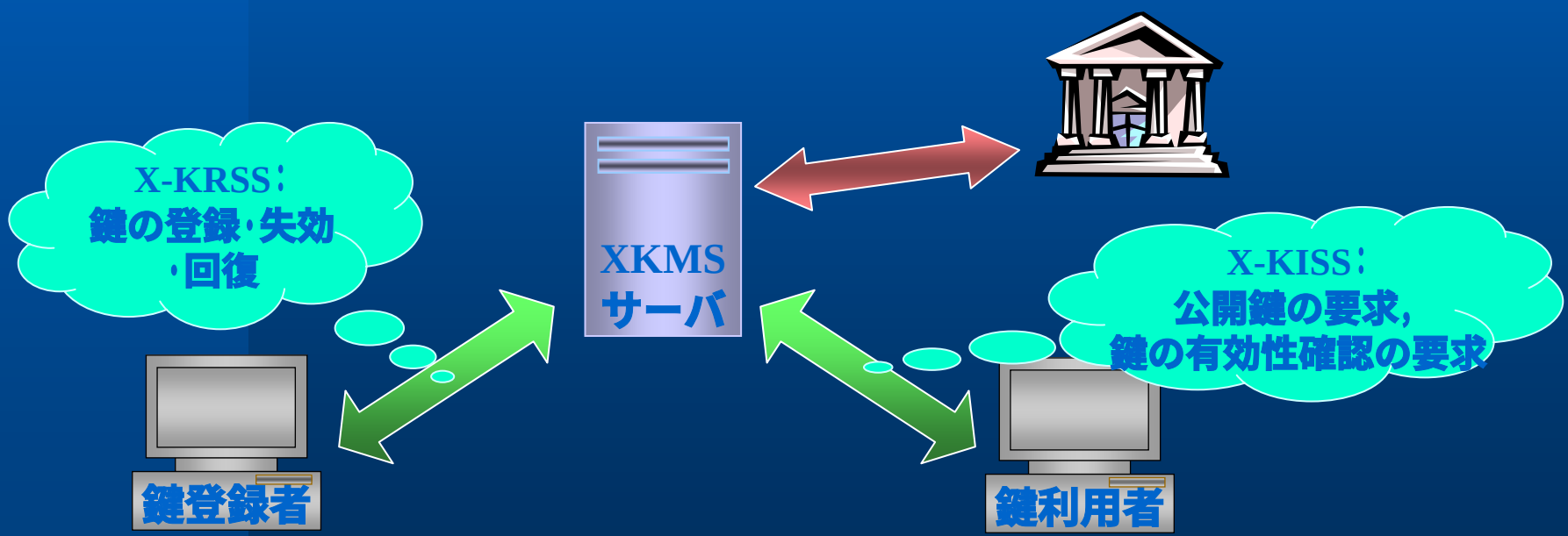
EncryptionMethod：
暗号ｱﾙｺﾞﾘｽﾞﾑ

EncryptedKey:
・コンテンツの暗号に使用した鍵（共通鍵）
・前記鍵を更に暗号化（公開鍵）

CipherValue：
暗号化されたコンテンツ

XKMS 2.0 (XML Key Management Specification)

- 公開鍵を登録･配布するメッセージとプロトコル
  - 公開鍵の登録 (K-KRSS)と問合せ (X-KISS)
  - 鍵利用者の処理を軽減する
  - 2002-03-18現在作業草案



XKMSのメッセージ (X-KISS)

＜Locate＞

- **サービスから公開鍵関連情報 (鍵値、証明書) を取得する**
- **例えば、暗号化の際に受信者の名前から受信者の公開鍵を取得する**
- **公開鍵関連情報はXML Signatureの<KeyInfo>を使用する**

```
<Locate xmlns="http://www.xkms.org/schema/xkms-2001-01-20">
  <Query>
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <KeyName>Encryption@beginnercard.com#RSAKey</KeyName>
    </KeyInfo>
  </Query>

  <Respond>
    <string>KeyName</string>
    <string>KeyValue</string>
  </Respond>
</Locate>
```

Query：公開鍵関連情報

Respond：問合せ項目

XKMSのメッセージ (X-KISS)

＜Validate＞

- **サービスに対して、公開鍵関連情報の有効性を問合せる**
- **例えば、署名検証の際に送信者が添付した公開鍵の有効性を検証する**

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

  <Respond>
    <string>KeyValue </string>
    <string>X509Data </string>
  </Respond>
</Validate>
```

Query：
公開鍵関連情報

Respond：
問合せ項目

XKMSのメッセージ (X-KISS)

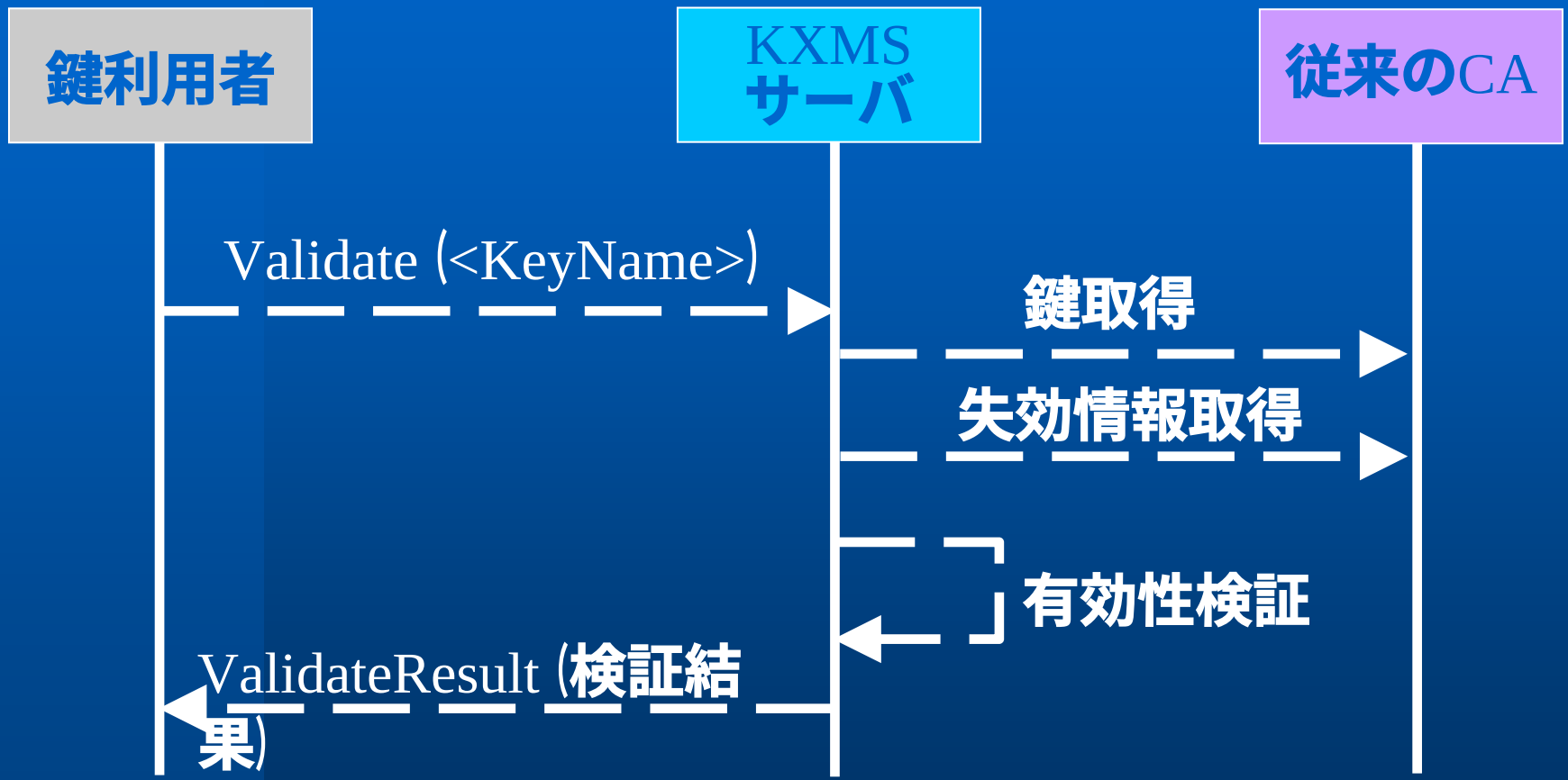
<ValidateResult>

```
<ValidateResult xmlns="http://www.xkms.org/schema/xkms-2001-01-20">
  <Reslut>Success</Reslut>
  <Answer>
    <KeyBinding>
      <Status>Valid</Status>
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <KeyValue>
          <RSAKeyValue>
            <Modulus>y0eZi+pL544O0anaCbHOF==</Modulus>
            <Exponent>AQAB</Exponent>
          </RSAKeyValue>
        </KeyValue>
        <X509Data>
          (証明書の詳細データ)
        </X590Data>
      </KeyInfo>
    </KeyBinding>
  </Answer>
</ValidateResult>
```

Answer：
問合せ結果

Update

XKMSのプロトコル



Validate時のプロトコル概要
(XKMSサーバ＋従来CAの構成)

ケーススタディ「旅行:予約」

- システム要件

<申込み>
阿波踊り××
<Card>
暗号化
</Card>
</申込み>

XKMS

JFBに同じ

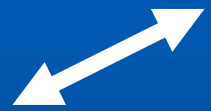
下田さん

XML Signature	XML Signature	XML Signature
XKMS	XKMS	XKMS
XML Encryption		XML Encryption

ケーススタディ「旅行:予約」

- 下田さん

<申込み>
阿波踊り××
<Card>
暗号化
</Card>
</申込み>

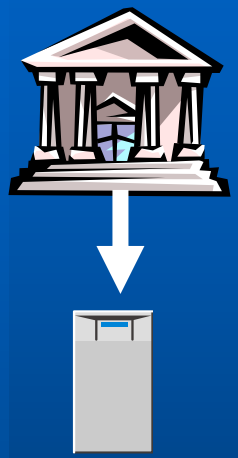


- 公開鍵の登録 (一回)
  - 鍵ペアの生成
  - XKMS Register：公開鍵を登録
- 暗号化
  - XKMS Locate：BEGINNERカードの公開鍵を取得
  - XML Encryption：暗号
- 電子署名生成
  - XML Signature：私有鍵で署名

XML Signature
XKMS
XML Encryption

ケーススタディ「旅行:予約」

- JFB



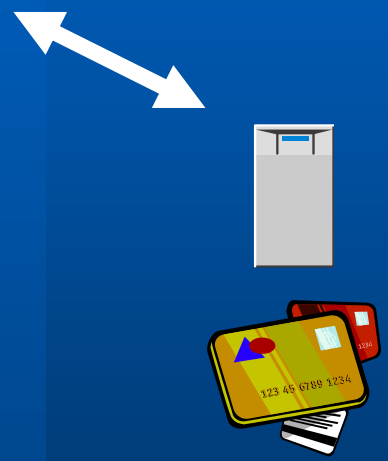
XML Signature

XKMS

- 電子署名検証
  - 下田さんの文書から公開鍵関連情報を取得
  - XKMS Validate：公開鍵関連情報の検証（同時に公開鍵の取得）
  - XML Signature：取得した公開鍵で下田さんの文書の署名を検証

ケーススタディ「旅行:予約」

- BEGINNERカード



- 公開鍵の登録（一回）
  - 鍵ペアの生成
  - XKMS Register：公開鍵を登録
- 電子署名検証
  （JFBに同じ）
- 復号
  - XML Encryption：自分の私有鍵で暗号部分を復号

ケーススタディ「旅行:予約」

- 結果
  - 下田さんからの予約はＪＦＢに安全に届き、予約は企画進行台帳に、個人情報は旅行者台帳に登録された。
  - クレジットカード情報はＪＦＢに知られること無く旅行者台帳に記録されている。（後日、ＢＥＧＩＮＮＥＲカード社に送付される。）
  - 同時に、ＪＵＳＴ航空・ホテルＳＩ2に対する手配が「仮予約」から「確保」に変更された。

ケーススタディ「旅行:確認」

3. 下田さんが予約を確認

   下田さんは、yasuitabi.comからの予約の確認メールに対して、「OK」の回答をする。

   JFB(yasuitabi.com)では、その回答をもとに手配を確保から本予約にし、手数料をクレジットカードにチャージする。

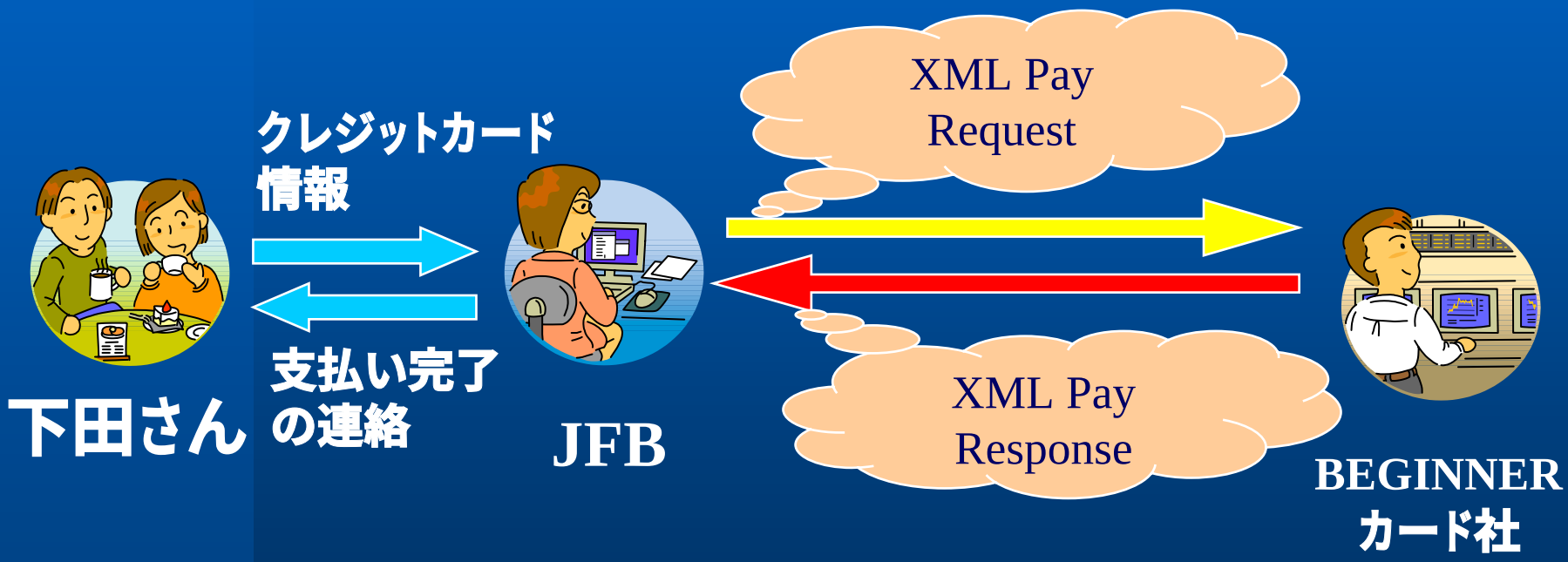
   JUST航空とホテルSI2は、それぞれクーポンを下田さんにeチケットとしてメールする。

ケーススタディ「旅行:支払い」

4. 下田さんが旅行会社に、旅行代金をクレジットカードで支払う
   - クレジットカード情報をJFBへ送る
   - JFB社はBEGINNERカード社へ決済要求をする
   - BEGINNERカード社は下田さんの与信情報を確認して決済結果をJFBへ返答する
   - JFBは支払いが完了したことを下田さんへ伝える

ケーススタディ「旅行:支払い」

- ケーススタディでの処理モデル



XML Pay仕様

(1) XML Pay: core

B2CとB2Bの支払い処理を統一するために必要な、XMLデータタイプを定義します。

(2) XML Pay: Registration

小売店の登録や構成のような、支払いに関する登録機能の自動化を定義します。

(3) XML Pay: Reports

バックオフィスで小売店の処理を報告する機能を、自動化するメカニズムを定義します。

XML Pay

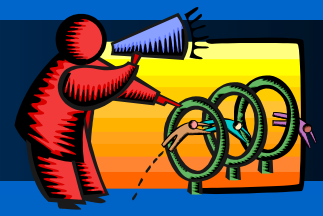
- **XML Payが目指すこと**

  “支払いを実行するために小売店サービスにサインアップすること”から、“支払い後に内容を報告すること”までの仕様を定めること。

- **XML Payとは**
  - **リクエスト(Request)**
  - **レスポンス(Response)**
  - **レシート(Receipt)**

  の3つの処理から成る。

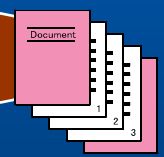
XML PayRequest(要求)



<XML PayRequest>

- <RequestData>
  - <MerchantId> — 小売店の識別ID
  - <Transactions> — 支払い取引リスト
- <RequestAuth> — 要求者の認証データ

JFB

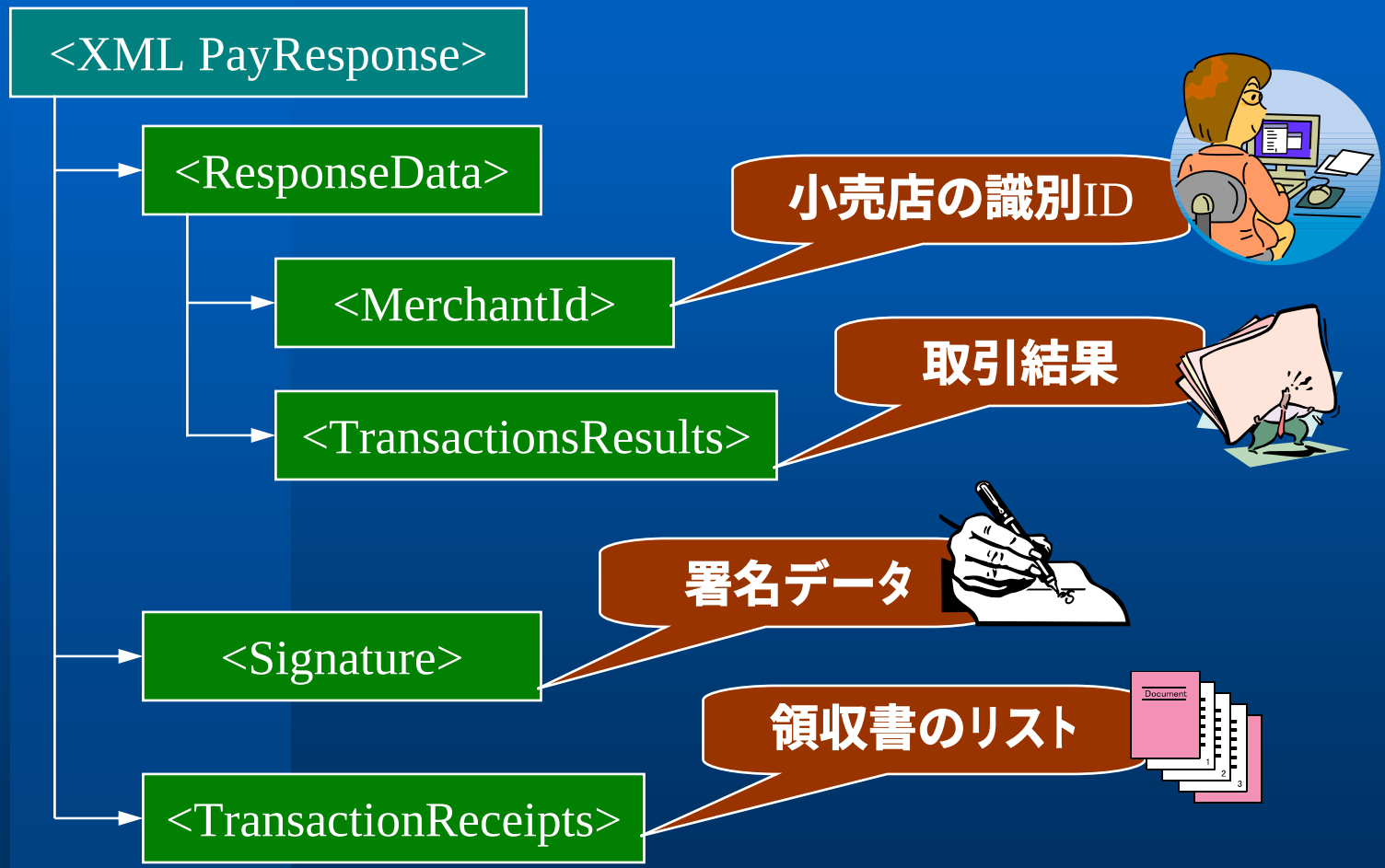


JUST航空：東京-徳島
往復2名分

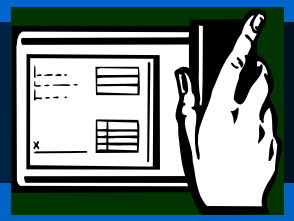
ホテルSI2：1泊2名

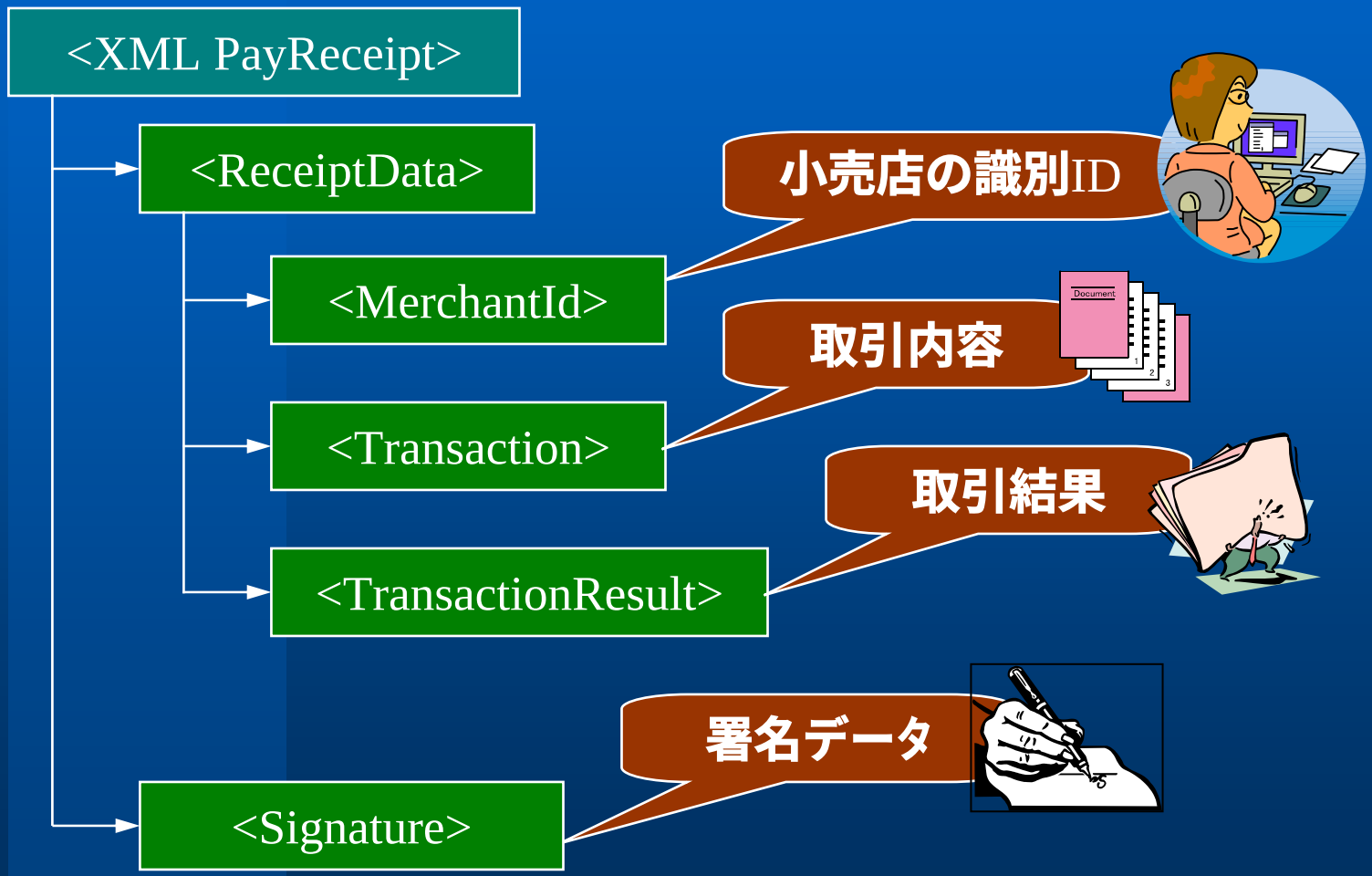
ＪＦＢの
ディジタル署名

XML PayResponse(返答)



XML PayReceipt(領収)





XMLPayRequest Document (1)

```
<Invoice>
 <BillTo>
  <Name>Shimoda</Name>
  <Address>Tokyo</Address>
  <EMail>shimoda@o-camera.com</EMail>
 </BillTo>
 <Items>
  <ItemNumber="1">
   <Description>AirFare</Description>
   <Quantity>2</Quantity>
   <TotalAmt>80000</TotalAmt>
  </Item>
  <ItemNumber="2">
   <Description>AccommodationCharges</Description>
   <Quantity>2</Quantity>
   <TotalAmt>35000</TotalAmt>
  </Item>
 </Items>
 <TotalAmt>115000</TotalAmt>
</Invoice>
```

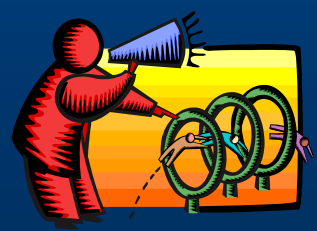
購入者情報
東京在住の下田さん

購入内訳1
羽田-徳島往復
航空券2名分

購入内訳2
SI2ホテル宿泊料2名

購入総合計
115000円

購入詳細



XMLPayRequest Document (2)

```
<Tender>
 <Card>
  <CardType>BEGINNER</CardType>
  <CardNum>1234567890</CardNum>
  <ExpDate>200207</ExpDate>
  <NameOnCard>Shimoda</NameOnCard>
 </Card>
</Tender>
```

カード会社名

カード番号

カード有効期限

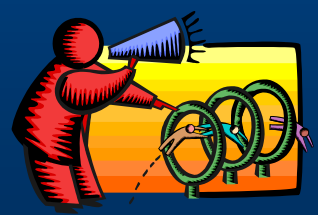
カード所有者名

クレジットカード情報を記述

```
<XMLPayRequest>
 <RequestData>
  <Transactions>
   (取引内容)
  </Transactions>
 </RequestData>
 (Signature)
</XMLPayRequest>
```

リクエストデータ

署名 (Option)



XMLPayResponse Document

```
<XMLPayResponse>
 <TransactionResult>
  <Result>0</Result>

  <AVSResult>
   <StreetMatch>match</StreetMatch>
   <ZipMatch>match</ZipMatch>
  </AVSResult>

  <CVResult>match</CVResult>

  <Message>AuthorizationApproved</Message>

  <PNRef>PN123412345</PNRef>

  <AuthCode>12345678</AuthCode>

 </TransactionResult>
</XMLPayResponse>

(Signature)
```

- 承認結果(0=承認)
- AVS(アドレス立証サービス)承認結果
- CV(Credit Void) 承認結果
- 結果の説明
- 支払い処理上の識別子
- 銀行による 取引認可コード
- 署名 (Option)

取引結果

XMLPayReceipt Document

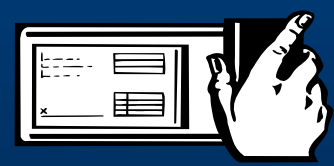
```
<XMLPayReceipt>
 <ReceiptData>
  <Transaction>……</Transaction>
  <TransactionResult>……</TransactionResult>
 </ReceiptData>
 (Signature)
</XMLPayReceipt>
```

取引内容

取引結果

受領データ

署名 (Option)



ケーススタディ「旅行:旅行」

5. 下田さんが友人と旅行

下田さんは友人とともに計画どおりに旅行する。

JUST航空とホテルSI2は、進行状況に応じて、JFB内の企画進行台帳を更新する。

JFBでは企画進行台帳の更新状況を確認し、旅行が問題なく進行しているかどうかをチェックする。

おしまい.....

ケーススタディ「旅行」

CAUTION:

ここでとりあげたモデルに登場する人物/団体等の名称は、すべてフィクションであり、実在する人物/団体等とはいっさい関係ありません

図解XML規格の概略説明

図解XML規格（セキュリティ編）第2版を参照ください。

まとめ

- セキュリティのXML規格は、インターネットでの相互接続に焦点を当てています
- 基礎技術の規格は普及中です
- セキュリティのXML規格は、よりアプリケーションに近いレベルのものへ焦点が移ります (例えば、権利保護・生体認証・再配置)

課題

- 規格化
  - 不確定要素（アプリケーション依存の定義等）は極力排除が望まれる
  - 規格化までの時間が長期化
    （XML Signature：要求仕様は1999年10月）
- 相互接続性
  - オプションの機能（アルゴリズム等）は使用しない
  - 普及が進む基盤技術の互換性を維持した機能拡張
  - W3C公開のExample、Test Resultを有効利用
- オープンソースコミュニティとの良好な関係
  - ソースコード公開による完全性の保証