XKMS

XKMSシステムの開発

XKMSサービスと、Java&.NETによるクライアント

2002年6月10日

XML Consortium 応用技術部会

セキュリティWG

本日の報告内容

- システム全体の説明
  - 背景と目標
  - 概要
  - システムの構成
  - 計画と実績
- XKMSサービス
- Java版XKMSクライアント
- .NET版XKMSクライアント
- デモンストレーション

メンバ紹介

- 富士ゼロックス (株)　　道村 唯夫
- 沖電気工業 (株)　　池上 勝美
- リコーシステム開発 (株)　小堀 真義

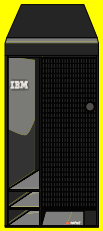
背景と目標

- 背景
  - 昨年度、Java Consortium XML部会において、電子署名に関する調査と、標準APIの提案を行った
    - 電子署名に関しては、大変理解が深まった
    - 市場においても、電子署名の実装がいろいろとでてきた
  - 電子署名の周辺/延長で、他の仕様を調査/検討してみたい
  - 鍵管理はセキュリティ(署名、暗号、など)において、たいへん重要であり、XKMSは有用であると認識している
  - XKMSのClient SDKが公開され、仕様としても固まりつつある

- 目標
  - XKMSサービスを提供できるサーバと、Java、.NETを使ったアプリケーションを実装し、相互接続を通じて XKMSを理解する

概要

鍵利用者



ASN.1処理
X.509証明書処理
パス生成
パス正当性確認

失効チェック
信頼モデル
PKCS/CMP/CMC/
CEP/CRS/LDAP/
OCSP

鍵保有者

公開鍵

私有鍵

鍵登録

ASN.1

Before XKMS

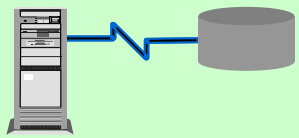
隠蔽

確認サーバ
(Locate &
Validation)

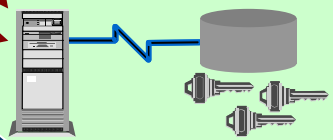
登録サーバ
(Registration)

XKMSを使うと…

? いろいろなサービス



ディレクトリ

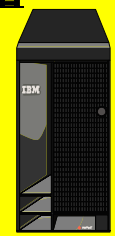


PKI Provider

鍵利用者

XML(X-KISS)

必要な公開鍵の要求



自分の公開鍵の登録要求

XML
(X-KRSS)

鍵保有者

公開鍵

私有鍵

After XKMS

詳細については…

他のプレゼンテーションや、OASISのサイトをご参照ください

システムの構成

- 今回実装したシステム

発注データの送受

Java版XKMSクライアント

.NET版XKMSクライアント

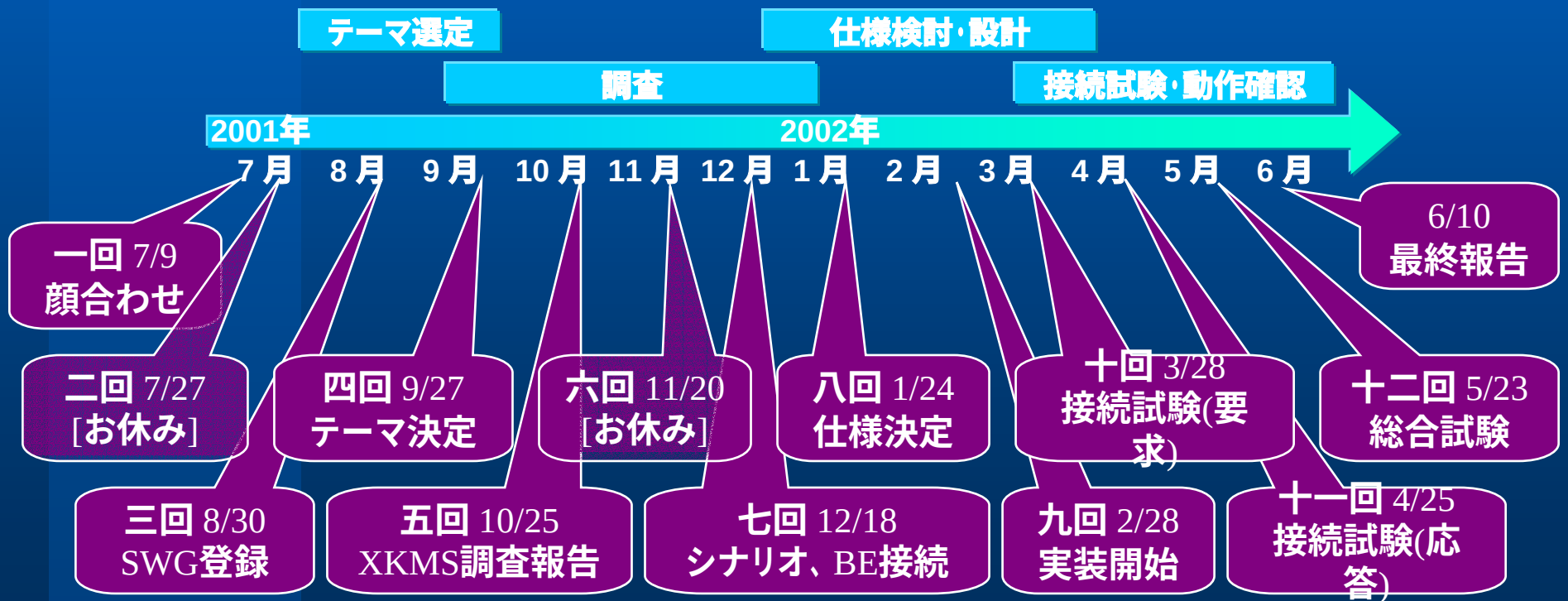
鍵の登録

鍵の検証

XKMSサービス

計画と実績

- 月に一回の応用部会のミーティングでの活動
  - 調査報告、経過･状況交換、接続･動作確認
- 調査や設計、実装は、自分で工夫した時間で



XKMSサービス

富士ゼロックス株式会社
道村 唯夫

XKMSサービス

- 報告内容
  - 作業実績
  - 設計と実装
  - 残作業
  - 考察と感想

作業実績

- 作業実績
  - 調査: 10H
    - XKMSの仕様の調査
    - Back End CAとの接続の調査
    - セキュリティ関連ツールの収集と動作検証
  - 設計: ??H
    - <<調査と実装の時間に含む>>
  - 実装と接続試験: 30H
    - テスト･ツールの作成
    - XKMSサービスの実装
    - Clientとの接続試験

設計と実装

- 目論んでいた仕様
  - 無料で利用できるCAをBack Endとする
    - Key Pairはサーバ側で生成する
  - Locate機能のRemoteReferenceは実装しない
  - Validate機能は実装しない
  - CRLの検証はしない
  - SSL/TLSは使わない
  - 署名の検証、付加は行わない
  - etc

XKMSサービス

既存認証局

設計と実装（続き）

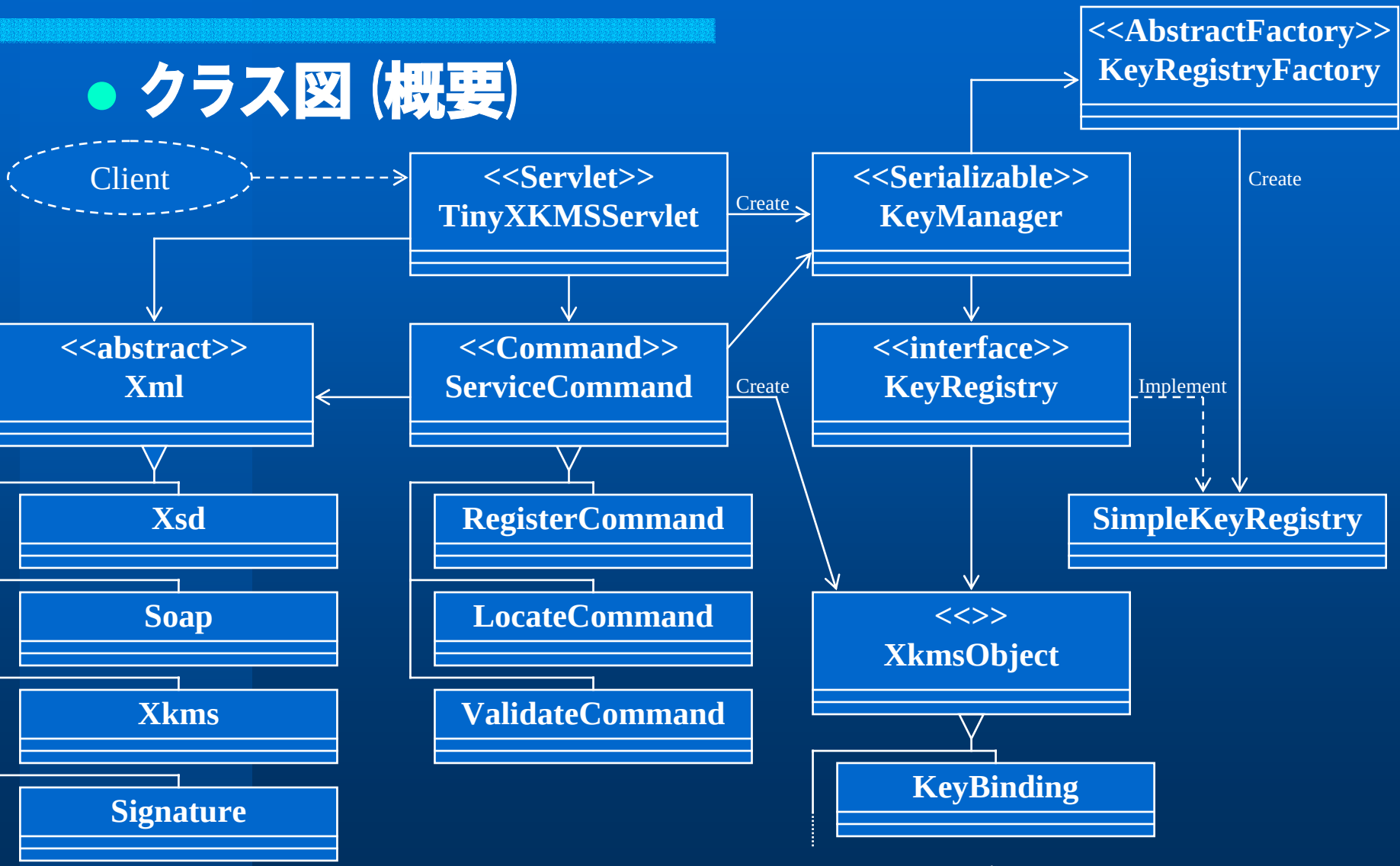
- 実装した仕様
  - Back End CAとの接続をあきらめ、簡易鍵管理機構を作成する
    - Key Pairはクライアント側で作成する
  - 署名の検証は行わないが、サーバからのレスポンスには署名を付加する
    - 付加する署名は、検証不可能なもの
- 実装規模
  - クラス数: 32（Exceptions、Messagesを含む）
  - ライン数: 2280（コメント等を含む)

設計と実装（続き）

- 環境
  - Windows 2000 Professional SP2
  - JDK 1.3.1
  - Tomcat 4.0.1 + Xerces-J
- 構成
  - XKMS Servlet
    - XKMSサービス本体
    - Key Registry Connection Plug-in Mechanism
  - Simple Key Registry
    - メモリ上で鍵情報を管理する簡易鍵レジストリ

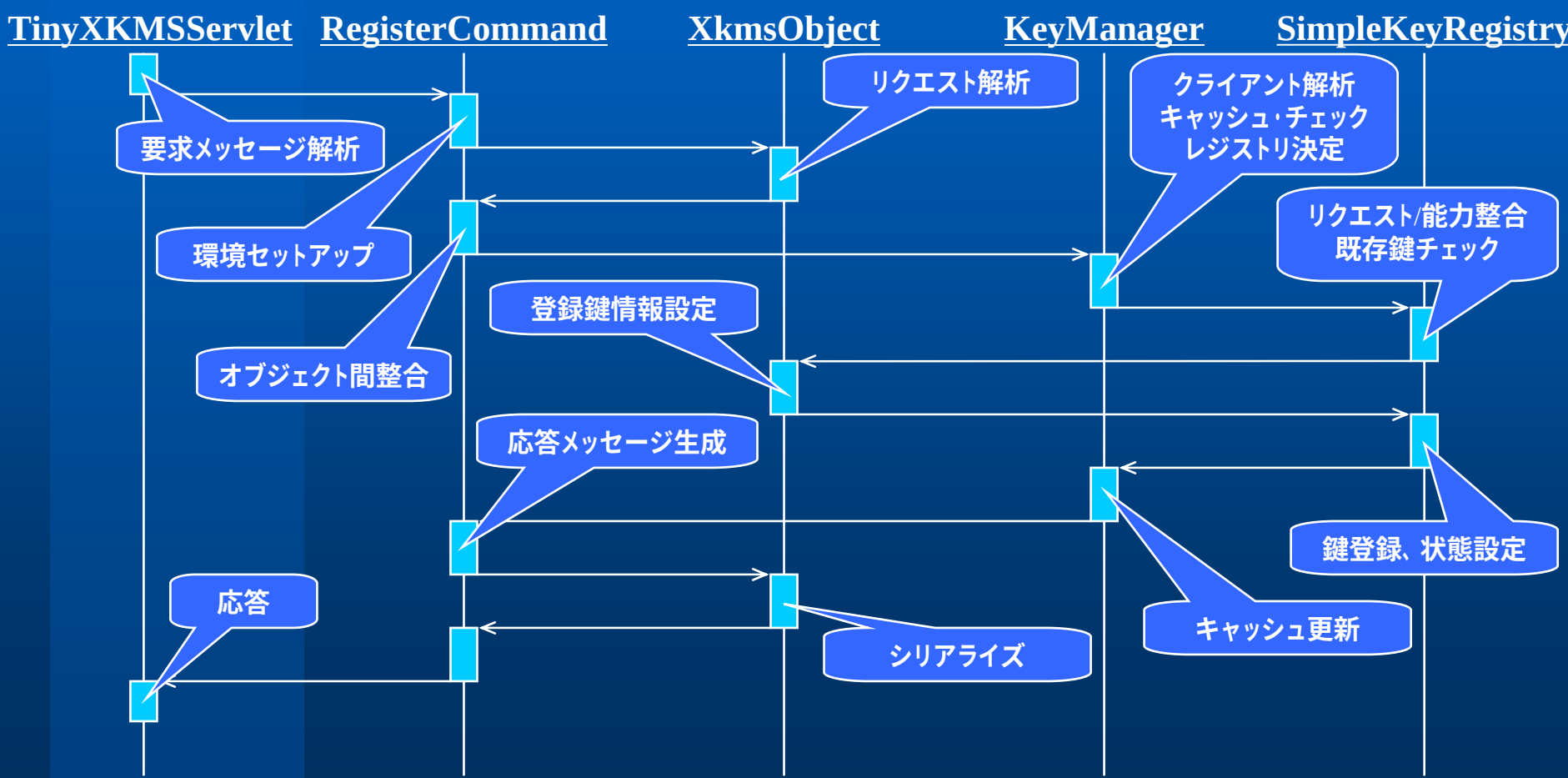
設計と実装（続き）

- クラス図（概要）



設計と実装（続き）

- シーケンス図（概要:Register/正常ケース）

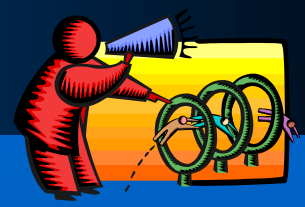


設計と実装（続き）

- KeyBinding Typeデータの使い方
  - XKMSにおいて最も重要なデータ要素
    - Request: Prototype、Response: KeyBinding

KeyBinding Type			Client	Server
TransactionID			Clientが自由に設定	指定されたものをそのまま返す
Status			要求内容の詳細を示す	鍵の状態を示す
KeyID			(使わない)	鍵を一意に決定するuri
XML Signature	KeyInfo		鍵の詳細データ	←
		KeyName	鍵を一意に決定する文字列	←
		KeyValue	公開鍵の値	←
		RetrievalMethod	鍵情報の取得方法と種別	←
		X509Data	X.509v3証明書データ	←
		PGPData	PGP鍵データ	←
		SPKIData	SPKI公開鍵関連データ	←
		MgmtData	In-band鍵配布、相互合意データ	←
PassPhrase			処理に必要なパスフレーズ	処理要求の許可･拒否を決定
ProcessInfo			(当事者間で決定)	←
ValidityInterval			鍵の有効期間を示す	←
KeyUsage			鍵の使用目的を示す	←

設計と実装（続き）



- 障害となった点、工夫した点など
  - SOAP over HTTPなのにSOAPActionヘッダなし
    - SOAP 1.1のメッセージとしては不適格
    - AxisのContainerは、メッセージを受け入れない

    → ServletでXMLのパースを実施し、CommandにはDOMのオブジェクトで要求メッセージを渡すよう変更
  - Content-Typeがapplication/xmlとtext/xmlの併記
    - これは正しい「media-type」ではない (cf. RFC 2616)

    → */xmlであれば受け入れるように変更
    SOAP 1.2では「application/soap」が正しい

設計と実装（続き）

- 障害となった点、工夫した点など（続き）
  - 応答メッセージには署名がないとエラーになる
    - メッセージへの署名付与はオプションであるにもかかわらず、Client SDKでは署名の検証が実施される

    → XKMSの応答メッセージに署名を付与するように変更
  - 処理結果の通知方法が複数存在する
    - HTTPレベル、SOAPレベル、XKMSメッセージレベル

    → 処理のステージに応じて、通知の方法を選択
  - 応答メッセージの国際化対応
    - メッセージ･リソースの追加により、各国の言語での応答が可能（要求メッセージ中のロケール情報をもとにロケールを切り替える）

残作業

- Back End CAへの接続
  - CAのサポートするプロトコル、証明書への対応
  - CRLステータスの確認
- 自己完結（CAを使わない）サービスとしての機能
  - 証明書生成
  - 鍵の生成と保存
- Passphraseへの対応
- 正しく署名に対応
  - 検証と生成
- SSL/TLSでの通信
  - Java/Tomcatの機能の利用 (Keytool,Server.xml)

考察と感想

- クライアントに対する利点が明確
  - 複雑な文法、表記法の理解が不要
    - X509証明書、ASN.1、…
  - 様々なプロトコルの知識が不要
    - LDAP、OSCP、…
  - CRLステータスの確認が不要
  - 信頼の輪(樹)に対する考慮が不要
    - クライアントにとっては、信頼できるXKMSサービスを利用することが重要

考察と感想（続き）

- 本仕様だけでは独立サービスの提供は不可能
  - そもそも、out-of-band-methodという記述がある
  - 現仕様だとCAと独立して運用することは無理
    - 仕様上の謎がある
    - 全ての手続きをオンライン化できない
  - CAやクライアントとの信頼関係の構築が必要
    - CAに対してクライアントの代理人として認証してもらう
    - クライアントに対して提供する情報の信頼度を向上
  - 鍵の保有者と利用者で同じサービスの参照が必要
  - サービスに関連する全員での事前合意が必須

考察と感想（続き）

- 仕様上の謎、問題点
  - 鍵の登録要求者の本人確認
    - X-KRSS利用者の本人確認の方法が存在しない
    - KeyNameとPassphraseの安全な割り当てが必要
  - KeyID
    - 形式的にuriであれば何でもよいが、使用目的が明確になっていないので、当事者間での事前合意が必要
    - Query時のKeyIDの使用方法が曖昧
  - KeyName
    - 一意性を誰が保証するのかが明確でない
    - VeriSignのサーバはクライアントの指定を上書きする

考察と感想（続き）

- 仕様上の謎、問題点（続き）
  - Passphrase
    - Passphraseによる認証は共有鍵＋Hashに基づく
    - この共有鍵をどうやって交換するのか?
  - サーバが応答する鍵情報の数の上限
    - WSDL上では、ひとつの鍵のRegister要求に対して、鍵情報がmaxOccurs=“unbound”となっている
    - Validate要求でも同様
  - KeyValueの種別
    - サーバでKey Pairを生成するRegister要求において、生成する鍵の種別を指定できない

考察と感想（続き）

- 仕様上の謎、問題点（続き）
  - Queryの文法
    - Query Elementのどの子要素が検索キーになり得るのかが不明確
    - そのQueryでの検索方法が不明(完全一致のみ?)
  - 証明書作成データの受け渡し
    - XKMS上の要素だけでは証明書を作成できない
  - ResultCode: Pending時の振る舞い
    - X-KISS要求の結果として、Pending(処理キューに入った)としたとき、サーバ/クライアントのすべきこと?
      - サーバ: 処理終了後に通知する方法がない
      - クライアント: 処理の結果を問い合わせる方法がない

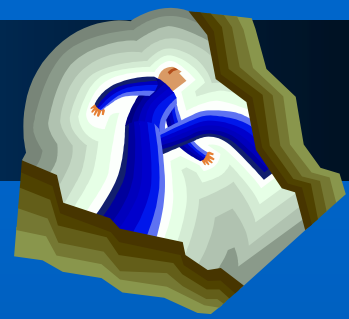
考察と感想（続き）

- 仕様に対する改善提案
  - より高い目標の設定
    - Back End CAのさらなる隠蔽
    - XKMSサービス間での協調
  - KeyNameの構造化と一元管理
    - Back End CAがわかるようにする(サービスのため)
    - 一意性の保証
  - KeyIDの有効利用
  - Passphraseの有効利用
    - Passphrase交換メッセージの定義
  - 登録要求者の本人確認方法

Java版XKMSクライアント

リコーシステム開発株式会社
小堀 真義

Java版XKMSクライアント

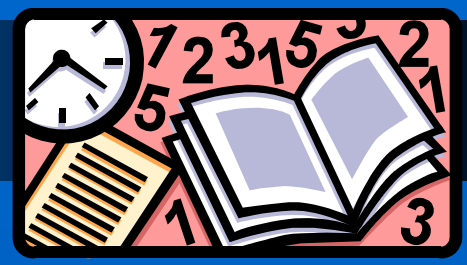


- 報告内容
  - 作業実績
  - 設計と実装
  - 残作業
  - 考察と感想

作業実績

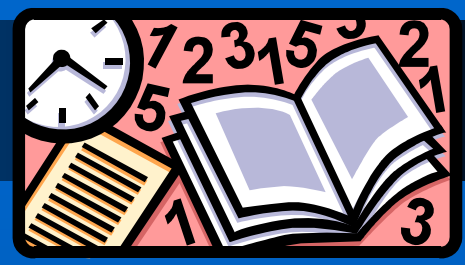
- 調査：約２０Ｈ
  - XKMSの調査
  - ＴSIKの調査
- 設計と実装：約２０Ｈ
  - VeriSignのテスト用サーバーとの接続
  - Java版クライアントの作成
  - ＸＫＭＳサービス、.NET版クライアントとの接続試験

設計と実装



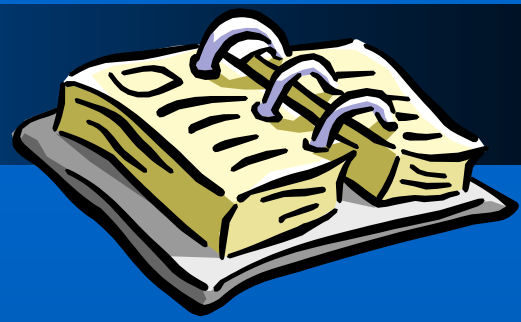
- 概要
  - 署名に必要な鍵を生成し、XKMSサービス経由でCAに登録する
  - 登録した鍵で署名を付加する

設計と実装(続き)



- 当初考えていた仕様
  - XKMSのRegister機能による鍵登録を実装する
  - Key Pairはサーバ側で生成する
  - XML Signatureによる署名の付加を実装する
  - Webアプリケーションとする
  - TSIKを使用する

設計と実装（続き）



- TSIKとは…
  - Trust Service Integration Kit の略
  - VeriSignが提供するクライアント側のセキュリティ・アプリケーション開発キットである
  - XKMS、XML Signature、XML Encryption等をインプリメントしている
  - Version 1.0（2002/06現在）
- 今回TSIKを用いて実装した機能
  - XKMSのRegister
  - XML SignatureのSign

設計と実装（続き）

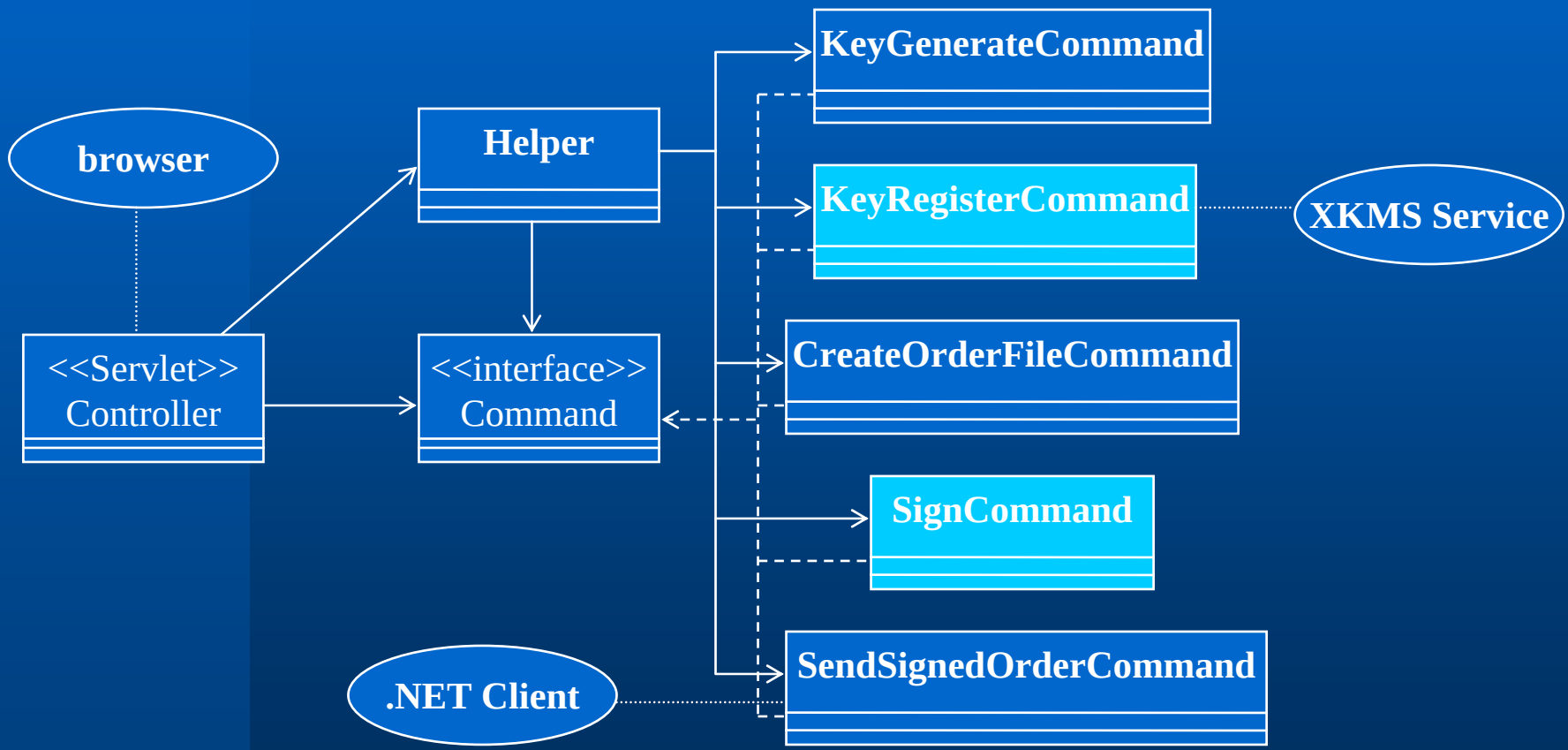
- 実装した仕様
  - Key Pairはクライアント側で生成
    - TSIKのドキュメントに、サーバ側での生成をサポートしていないという記載があったため
    - しかし、実際は実装されているようである
- 規模（2002/05/30現在 当日までに修正）
  - ライン数：約1000ライン（コメント等含む）
  - クラス数：12
  - その他にHTML、JSP、JavaBean

設計と実装（続き）

- 環境
  - Windows 2000 Professional SP2
  - J2SDK 1.4.0
  - Jakarta Tomcat 4.0.2
  - ＴＳＩＫ 1.0

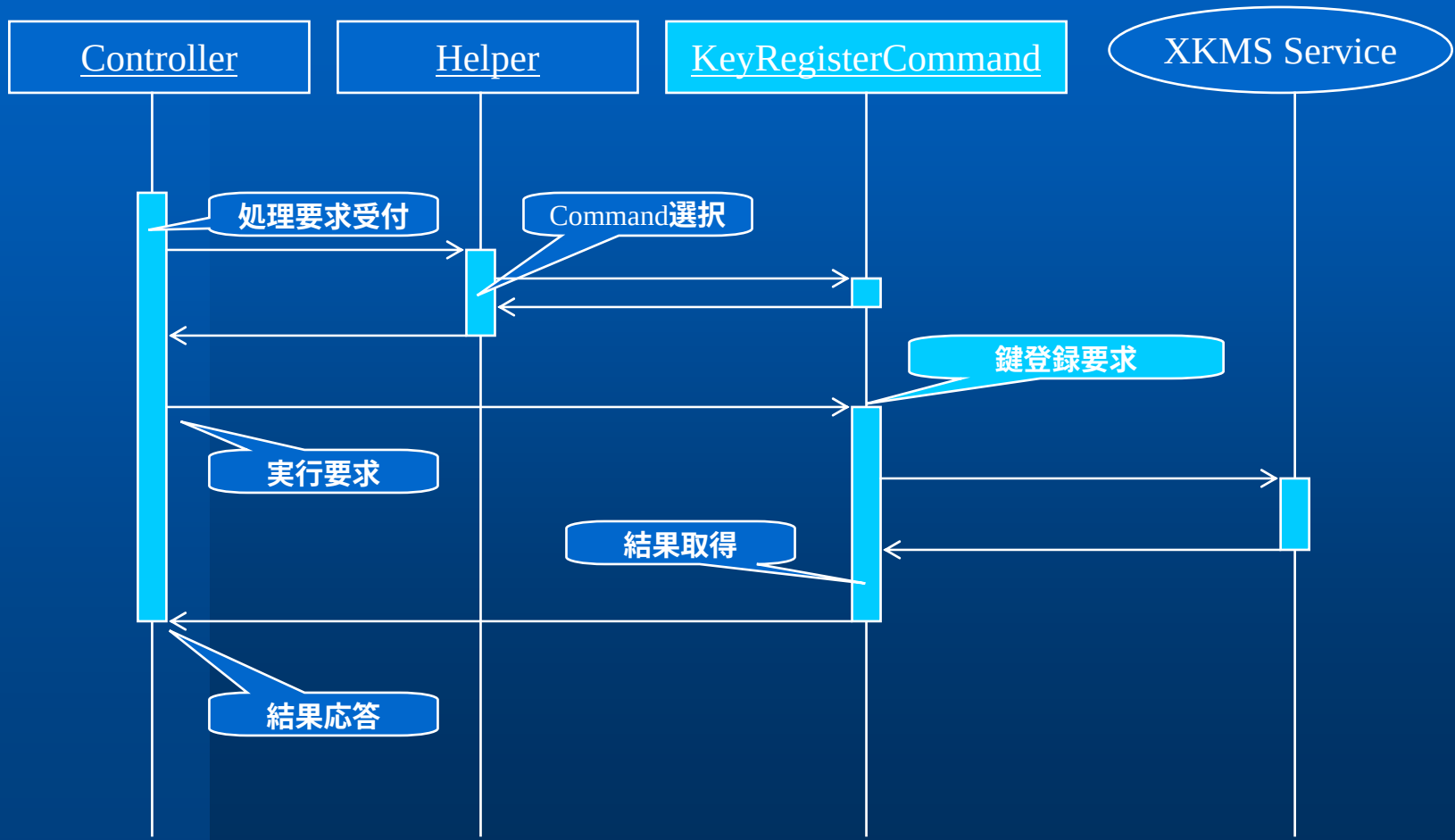
設計と実装 (続き)

- クラス図 (概要)



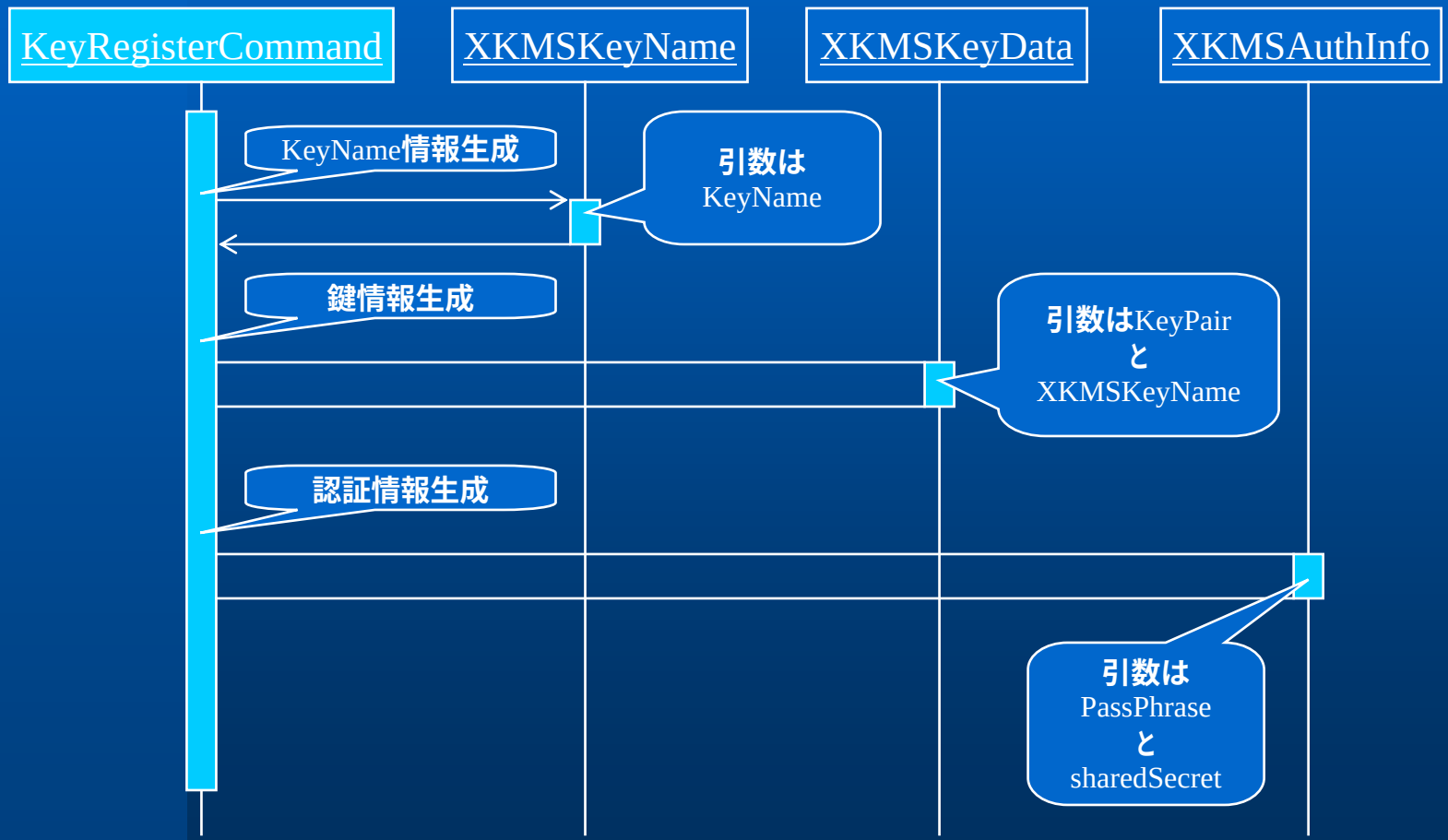
設計と実装（続き）

- シーケンス図（概要：Register/正常ケース）



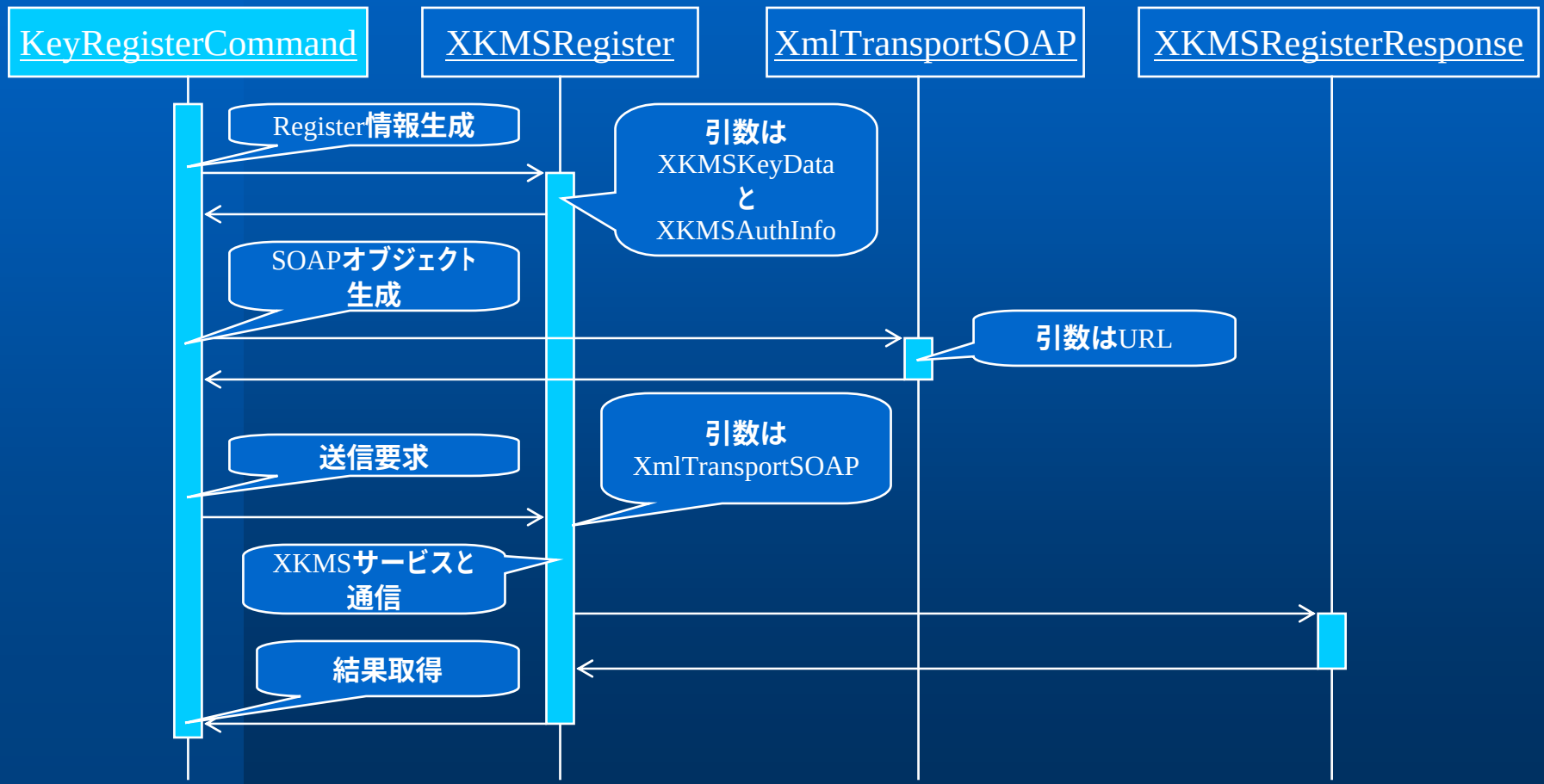
設計と実装（続き）

- シーケンス図（Register/TSIK使用部分①）



設計と実装（続き）

- シーケンス図（Register/TSIK使用部分②）



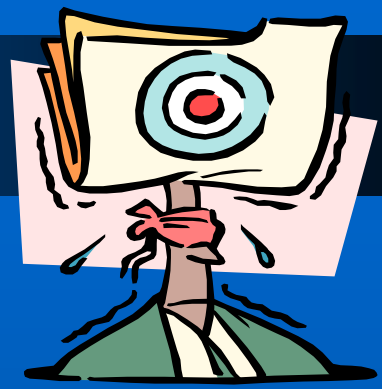
設計と実装（続き）

- Register/TSIK部分ソースコード

```
XKMSKeyName  keyName = new XKMSKeyName ( name );
XKMSKeyData  data = new XKMSKeyData ( key, keyName );
XKMSAuthInfo  authInfo = new XKMSAuthInfo ( passPhrase, sharedSecret );
XKMSRegister  register = new XKMSRegister ( data, authInfo );
XmlTransportSOAP  soap = new XmlTransportSOAP (  url );
XKMSRegisterResponse resp = register.sendRequest ( soap );
```

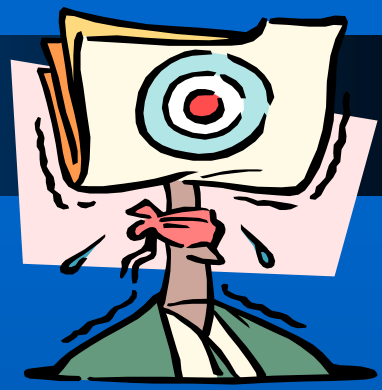
  - 単純な実装で済む
  - Locate、Validate、Revokeも同程度の実装量

設計と実装（続き）



- 障害となった点
  - HTTPヘッダに「SOAPAction」が付加されない
    - 仕様では必須となっているが、TSIKでは付加されない
    - XKMSサービス側で対応してもらうことにより解決
  - XKMSサービスからのレスポンスには必ず署名が必要
    - 仕様では必須項目ではないが、ＴＳＩＫでは必須
    - XKMSサービス側でダミーの署名を付加したもらい解決

設計と実装（続き）



- 障害となった点（続き）
  - 「dsig:Reference」にXPointerが使用される
    - 仕様上は間違いではないが、.NET版クライアントでは対応していなかった
    - 旧バージョンのクラスを使用することにより解決

残作業

- Key Pairのサーバ側生成対応
  - TSIKには実装されているため、サーバ側が対応していれば可能
- 他の機能への対応
  - Locate、Validate、Revoke機能の実装

考察と感想

- XKMS
  - クライアントの開発が容易になる
    - X509、ＬＤＡＰといった難解な仕様の理解は不要
    - TSIKのような優れた開発キットがあればXKMSの理解もほとんど必要ない
  - 運用が容易
    - 鍵生成、登録をXKMSサービスが行う
    - 公開鍵の管理はＸＫＭＳサービスがすべて行う

PKIの垣根を低くすることができる

考察と感想 (続き)

- XKMS (続き)
  - 仕様が曖昧
    - KeyNameの扱いが明確でない、等
  - 現状ではXKMSサービスとクライアントはセットで開発する必要がある
    - 現状の曖昧な仕様では互換性に問題が発生する

考察と感想 (続き)

- TSIK
  - 洗練されたAPI
    - XKMSの仕様の大部分を隠蔽しているため、XKMS、XML Signatureの仕様をほとんど理解しなくとも実装が可能である
  - ログの充実
    - 標準でログ出力をサポートしているため、デバッグが容易である
  - 実装が容易
    - わかりやすいサンプルが付属している

考察と感想 (続き)

- TSIK (続き)
  - カスタマイズが困難
    - レスポンスの署名のチェックをはずすことが出来ない、dsig:ReferenceをXPointer以外の方法に変更できない、等
  - 互換性の問題
    - 仕様が曖昧なので仕方がない？
    - 限定した接続先を想定している？
    - XKMSの仕様に準拠していない部分がある

考察と感想（続き）

- Webサービスとしての実現可能性
  - 公開鍵、証明書の管理はすべてＸＫＭＳサービスに移譲することが可能
  - 秘密鍵の管理は依然としてローカルで行う必要があり、WebサービスではWebサーバ側で行うことが必要となるため、運用が難しい

考察と感想 (続き)

- 今後の発展
  - 仕様の明確化が必要
    - 互換性
    - 優れた開発ツールを促す意味でも
  - TSIKのような優れた開発ツールのさらなる登場
    - 短期間での開発を可能とする
    - より一層垣根を低くする

XKMSクライアント for．NET

～ 一から始めたWebサービスクライアント（初級編） ～

沖電気工業株式会社
池上 勝美

XKMSクライアント for．NET

- 報告内容
  - 背景と目標
  - 設計と実装
  - TIPs
  - 考察と感想

背景と目標

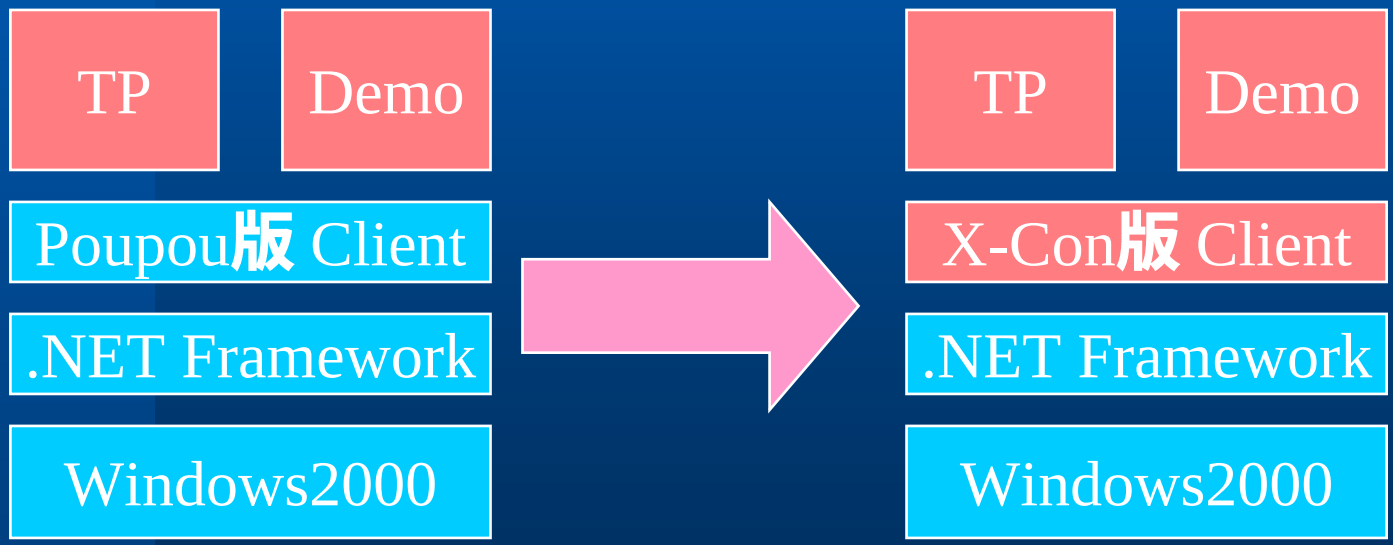
- 背景（全体背景に加え）
  - .NETでもVeriSignが提供するXKMSを利用できるか？
  - VeriSignが提供するTSIK（Java版）と電子署名の相互検証はできるか？
  - .NET版ClientSDKが利用できる
    （Poupou版：http://www24.brinkster.com/xkms）

- 目標
  - .NET版ClientSDKを利用したTP/Demoを作成し、VeriSign版Client、開発サービスへの接続を確認する
  - VeriSign版Clientとの電子署名の相互交換を確認する
  - 上記を通してXKMSの理解を深める

はずでしたが．．．．．

背景と目標(続き)

- Poupou版が動かない.....
- 追加の目標
  - X-Con版Clientも開発する



設計と実装

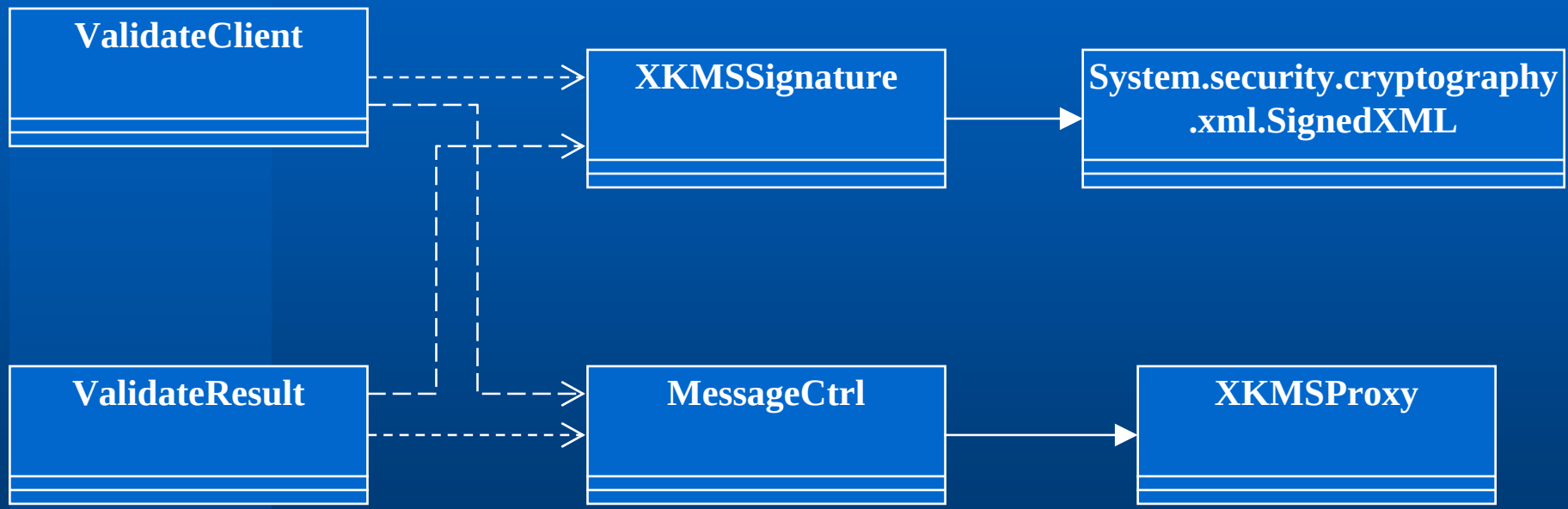
- 目論んでいた仕様
  - X-Con版Client
    - TP、Demoに対してXKMSの仕様を隠蔽したクラスを提供する（Poupou版相当）
  - TP
    - 同時に開発するサービス、及びVeriSignが提供するサービスに対してValidate、Locate、Registerのリクエスト発行とレスポンス取得ができる
  - Demo
    - TPの機能に加え、VeriSign版ClientとXML Signatureの相互検証ができる
  - etc

設計と実装（続き）

- 環境
  - Windows 2000 Professional SP2+SP3α
  - Microsoft Visual Studio .NET日本語版+SP1
- 構成
  - X-Con版ＸＫＭＳ Client for .NET
    - XKMS Client本体
  - TP
    - 各メッセージをカスタマイズして発行
    - レスポンスを取得
    - ログの取得
  - Demo（Demoでご紹介）

設計と実装（続き）

- クラス図（Validate概要）



TIPS

- つまづいた点
  - Proxyクラスの生成（Web Referenceの追加）
    - Proxyは生成されるがインスタンスのオートコンプリートが出来ない

      （http://xkms.verisign.com/xkms/Acceptor.nano）

      →
    - 参照したサービスのスキーマに誤りがあると、正常にProxyが生成されない（エラーも発生しない）
    - wsdl.exeでアラームが発生している

      .NET用のWSDL：
      http://www.xmltrustcenter.org/xkms/dotnet/resources/XMLKeyManagement.wsdl

TIPs（続き）

- スタブサービスの構築
  - ローカルデバック用にスタブサーバを構築したが、クライアントでインスタンス化に失敗する

  →wsdl.exe /Serverで生成されるのは<MustInherit>

- 署名検証のエラー（空白）
  - TSIKは署名生成後に空白で整形して出力する
  - .NETで検証すると失敗する

  →「*xmldoc*.PreserveWhitespace=False」でロード

- 署名検証のエラー（<Reference uri="XPointer"）
  - （VeriSign版Clientで報告済み）

残作業

- 仕様見直し
- リファクタ

考察と感想

- コツさえ掴めば.NETは簡単！
- クライアントプログラム
  - クライアントに複雑な処理が必要なWebサービスは、クライアントプログラムの配布が必要では？（Java版＋.NET版）

おち．．．．．

- MsdnからXKMSのサンプルコードが．．．．．
  - http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnaspp/html/implementingxkms.asp

Update

デモンストレーション

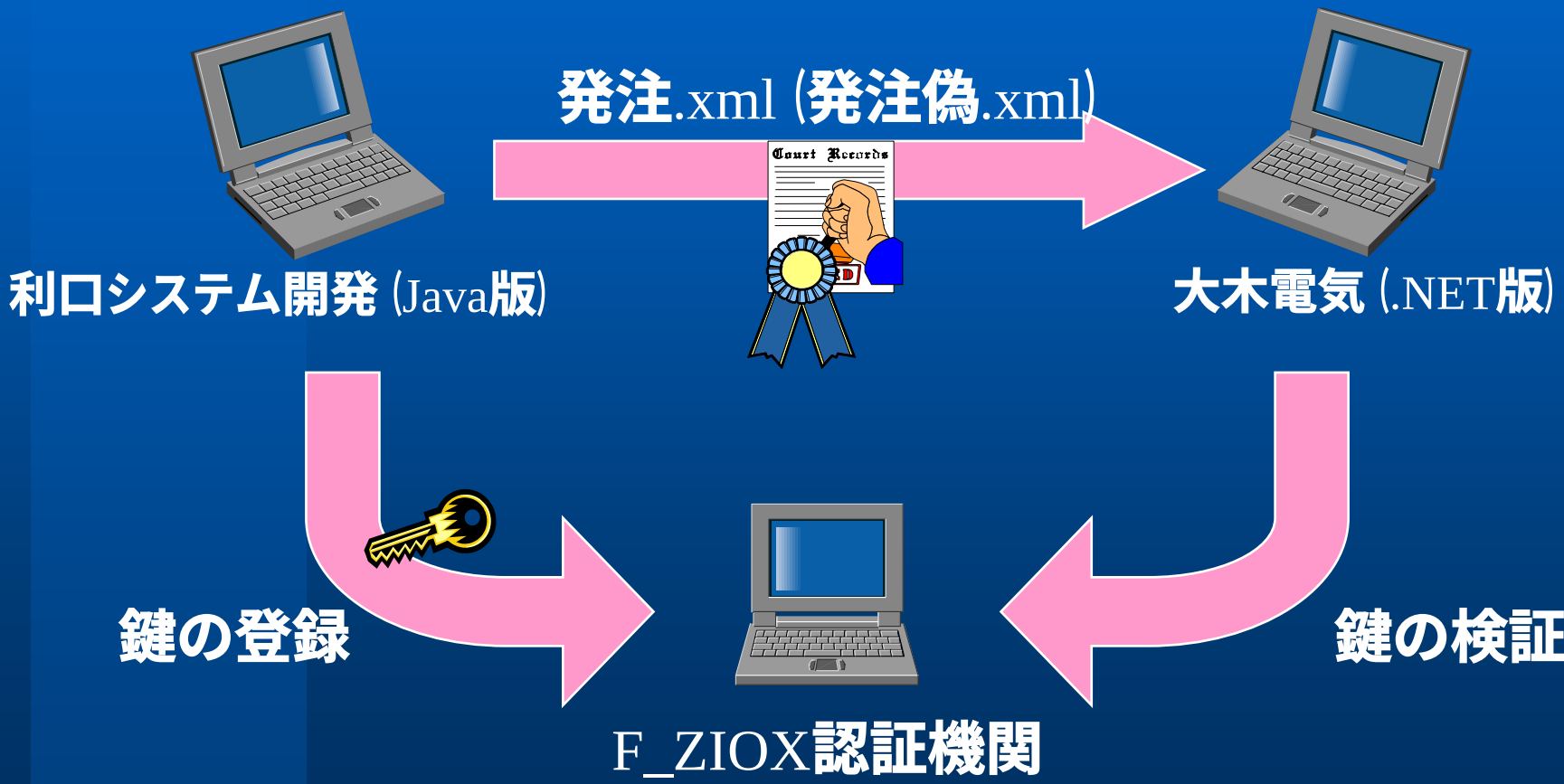
XML Consortium 応用技術部会
セキュリティWG

デモンストレーション

- あらすじ
  - 利口システム開発が大木電気に“GAIO_PC5X5”、増設メモリとマウスを発注します。
    - 発注書類 (“発注.xml”) には電子署名を付与します。
    - 運用に必用なPKIはF_ZIOX認証機関を使用します。
- ステップ
  - 正常系
    1. 利口システム開発は鍵ペアを生成し、公開鍵をF_ZIOXに登録
    2. 利口システム開発は“発注.xml”に署名を付与
    3. 大木電気は“発注.xml”の公開鍵をF_ZIOXで検証
    4. 大木電気は“発注.xml”の署名を検証
  - 異常系
    1. 登録しない公開鍵を使用して“発注偽.xml”に署名を付与

デモンストレーション (続き)

- デモシステム構成



デモンストレーション．．．．．

まとめ

- 活動成果
  - 一連の動作を検証可能なデモシステムが完成
  - 仕様を深く理解
- XKMSの仕様
  - XKMSはPKI普及の起爆剤になる
  - 現状の仕様は不明点が多数！
    （今後、W3Cでの議論に注目）
- その他
  - 複雑なサービスはクライアントプログラムも必要

２００２年度の活動案

- 調査
  - セキュリティ関連規格のウォッチング継続
- 実装
  - XKMSのフル実装（or 2.0対応）and Re-factor
  - XML Encryption
  - SAML ＋ XACML

セキュリティ関連XMLに興味のある方は、XMLコンソーシアムの
基盤技術部会
応用技術部会
のセキュリティSWGへ御参加ください

ご静聴ありがとうございました

RICOH SYSTEM KAIHATSU COMPANY, LTD.
リコーシステム開発株式会社

e. SolutionCore™

e.SolutionCoreを柱に、e-Solution、Security-Solution、
Image-Solutionを ご提供します。

http://www.rsk-tokyo.co.jp/

OKI
もっと広く、もっと早く、もっと確かに。
ネットワークソリューションの沖電気

電子署名サーバ
(開発中)

http://www.oki.com/jp/RDG/JIS/sas/index.html

カメラ記者クラブ 特別賞
CAMERA GRAND PRIX 2002

MINOLTA

世界で一番、薄い、小さい、そして軽い。
フルタイム・フラットの光学3倍ズームデジタルカメラ。

http://www.minolta.co.jp/japan/

THE DOCUMENT COMPANY
FUJI XEROX

SDES証明書発行サービス

SDES証明書発行サービスは、ASPまたは他の業務システムなどのセキュリティ基盤(PKI)の構築のための公開鍵証明書の発行機能をインターネットサービスとして提供いたします。

http://www.fujixerox.co.jp/product/cat/service.html

付録

- Acronyms

ASN.1
Abstract Syntax Notation 1

CA
Certification Authority

CEP
Certificate Enrollment Protocol

CMC
Certificate Management protocol using CMS

CMP
Certificate Management Protocol

CMS
Cryptographic Message Syntax

CRL
Certificate Revocation List

CRMF
Certificate Request Message Format

CRS
Certificate Request Syntax

DOM
Document Object Model

DSA
Digital Signature Algorithm

HTTP
Hyper Text Transfer Protocol

LDAP
Lightweight Directory Access Protocol

MAC
Message Authentication Code

OCSP
Online Certificate Status Protocol

PGP
Pretty Good Privacy

PKCS
Public Key Cryptography System

PKI
Public Key Infrastructure

付録（続き）

- Acronyms (cont.)

RSA
Rivest, Shamir, Adleman

SDK
Software Development Kit

SHA
Secure Hash Algorithm

SOAP
Simple Object Access Protocol

SPKI
Standard Public Key Infrastructure

SSL
Secure Socket Layer

TLS
Transport Level Security

WSDL
Web Service Description Language

WWW
World Wide Web

URI
Uniform Resource Identifier

URL
Uniform Resource Locator

URN
Uniform Resource Name

X-KISS
XML Key Information Service Specification

X-KRSS
XML Key Registration Management Specification

X509
ISO X.509 recommendation

XKMS
XML Key Management Specification

XML
Extensible Markup Language

付録（続き）

- Links
  - XKMS
    http://www.w3.org/TR/xkms/
    http://www.xmltrustcenter.org/xkms/
  - SOAP
    http://www.w3.org/2002/ws/
  - WSDL
    http://www.w3.org/TR/wsdl/
  - XML Signature
    http://www.w3.org/TR/xmldsig-core/
  - XML Encryption
    http://www.w3.org/TR/xmlenc-core/