Web Service Security - XKMS (TrustPoint)

Daniel Bär · Andreas Philipp · Norbert Pohlmann

Fachhochschule Gelsenkirchen
Fachbereich Informatik
Verteilte Systeme und Informationssicherheit
Neidenburger Str. 43, D - 45877 Gelsenkirchen
daniel.baer@informatik.fh-gelsenkirchen.de
norbert.pohlmann@informatik.fh-gelsenkirchen.de

Utimaco Safeware AG
Transaction Security
Germanusstrasse 4, D - 52080 Aachen
andreas.philipp@aachen.utimaco.de

Zusammenfassung

Web-Services sind erwachsen geworden und haben ein beachtliches Potential entwickelt: sie basieren auf einem offenen, dynamischen Austausch von Daten. Die Offenheit ist ihre Stärke und sorgte für eine breite Akzeptanz. Diese Offenheit ist aber mangels Sicherheit gleichzeitig auch der Hemmschuh, um Web Services auf breiter Basis in der Praxis einzusetzen. Web Services müssen sicher werden, um schützenswerte Daten angemessen übertragen zu können.
Voraussetzungen für den sicheren elektronischen Datenaustausch von Informationen über elektronische Wege sind Vertraulichkeit, Integrität und Verbindlichkeit. Verschlüsselung und digitale Signatur auf der Basis krytographischer Verfahren sind hierfür geeignete Verfahren. Eine Public Key Infrastructure (PKI) stellt hierfür geeignete Software, Protokolle und Standards bereit. Der richtige und langfristige Weg Web Services abzusichern, ist die Nutzung einer PKI. Der Aufbau und Betrieb einer PKI ist aber relativ aufwändig und bedarf verschiedener Protokolle auf der Client-Seite. Dies kann zu einer komplexen Angelegenheit führen, zu der nicht alle Anwendungsprogramme bzw. Anwendungsgeräte in der Lage sind.
Neue Ansätze ermöglichen eine einfache Kommunikation mit einer PKI. Web-Services und das Simple Object Access Protocol (SOAP) sind einfache Mittel, um in einer Service Orientated Architecture (SOA) entfernte Dienste in Anspruch zu nehmen. Die XML Key Management Specification (XKMS) definiert ein Protokoll, um Validierung und Verwaltung von Schlüsseln auf Basis von XML via Web Services zu verwirklichen. Die resultierenden Vorteile machen den Umgang mit einer PKI einfacher und schlanker. Die vorliegende Arbeit stellt die XKMS-Spezifikation vor, erläutert deren Funktionsweise, nennt Vor- und Nachteile und gibt Einblick in die Realisierung eines XKMS Responders im Rahmen des Projektes TrustPoint.

1 Web Services und IT-Sicherheit

Web Services erfüllen den Wunsch nach Interoperation heterogener Systeme - und das plattformneutral und herstellerunabhängig. Sie machen das möglich, was bis vor einigen Jahren nur eingeschränkt oder nur über komplizierte Mechanismen (beispielsweise Electronic Data Interchange, kurz: EDI) funktionierte: uneingeschränkter Austausch von IT-Applikationen und Diensten über standardisierte Verfahren zur gemeinsamen Nutzung und Verarbeitung von Daten.

Alles wunderbar? Wäre da nicht die Kehrseite der Medaille: Die bei Web Services verwendeten Datenaustauschverfahren - wie zum Beispiel das HTTP-Protokoll - sind zunächst einmal notwendigerweise offen. Dies birgt vielschichtige Sicherheitsprobleme von der Netzwerkebene bis hin zur Applikationsebene.

Bei Web Services handelt es sich um ein Bündel von Protokollen und Verfahren zum Austausch von Informationen. Dazu gehören Standards wie Universal Description, Discovery and Integration (UDDI), ein Registratur- und Verzeichnisdienst mit den dazugehörigen Extensible-Markup-Language-Schnittstellen (XML) oder die Web Services Description Language (WSDL), eine Sprache zur Beschreibung der Funktionen eines Web Services. Jedoch das Protokoll, das die meisten Web Services zum Nachrichtenaustausch benutzen, ist das Simple Object Access Protocol (SOAP), das über HTTP oder Simple Mail Transfer Protocol (SMTP) transportiert wird. Das klassische SOAP-Protokoll bietet keinerlei Sicherheitsfunktionalitäten wie Integritäts- oder Vertraulichkeitsschutz. Hier setzt nun die Sicherheit im Rahmen von Web Services auf. Im Rahmen des Methodenaufrufs per SOAP werden nun entsprechen Verfahren zum Schutz vor nicht berechtigtem Aufruf von Methoden sowie Verfahren zum Schutz der Vertraulichkeit, Integrität und Authentizität definiert. Die nachfolgende Grafik verdeutlicht noch einmal die bestehende Problematik, dass zum einen die Absicherung des Web Services an sich ein Problem darstellt (der im Rahmen des Artikel nicht behandelt wird) und zum anderen die Absicherung der Kommunikationsbeziehungen untereinander einen weiteren Themenkomplex darstellt.

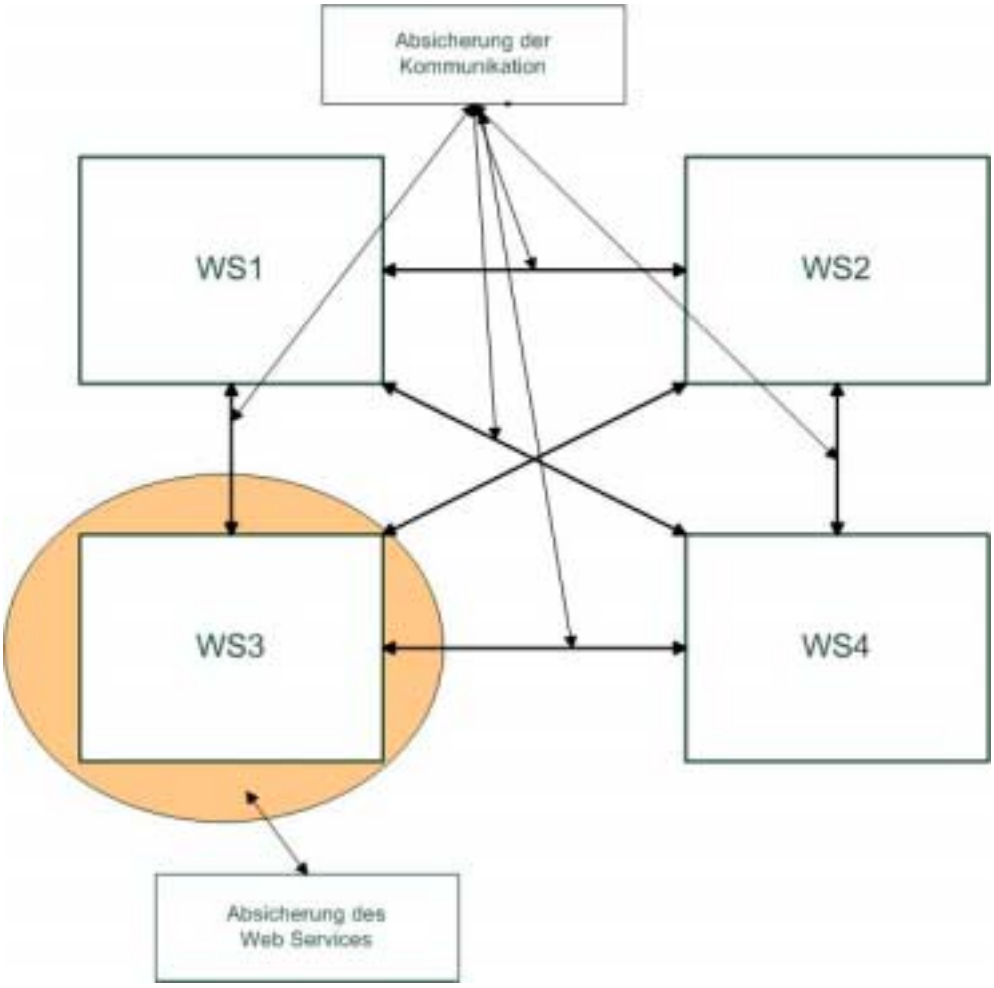

Abbildung 1: Schutz eines Web Services und Vertrauenswürdigkeit untereinander herstellen.

Durch den Standard WS-Security steht eine Spezifikation zur Verfügung mit deren Hilfe es nun möglich ist, eine durchgehende Sicherung der SOAP Nachrichten basierend auf Token, X.509 Zertifikaten sowie durch Einsatz von Verschlüsselungsverfahren zu realisieren. WS-Security definiert im ersten Schritt innerhalb eines XML-Dokuments einen Rahmen zur Einbettung der Sicherheitsinformation in eine SOAP-Nachricht. Die Wahrung der Integrität und Vertraulichkeit wird mit Hilfe der Sicherheitsstandards XML-Encryption und XML-Signature gewährleistet. Hiermit stehen nun zwei Verfahren zur Sicherung der Datenintegrität und Urheberschaft sowie zur Gewährleistung der Vertraulichkeit zur Verfügung.

2 Probleme mit PKIs

Der Aufbau und der Betrieb einer PKI Struktur ist komplex. Hinzu kommen hohe Investitions- und Integrationskosten. Des Weiteren ist bei einer PKI, die nicht nur unternehmensinternen Nutzen haben soll, die Zusammenarbeit und Interoperabilität mit externen PKIs notwendig. Die nachfolgende Grafik veranschaulicht noch einmal die wichtigsten Aspekte hinsichtlich der Kommunikations- und Protokollvielfalt.

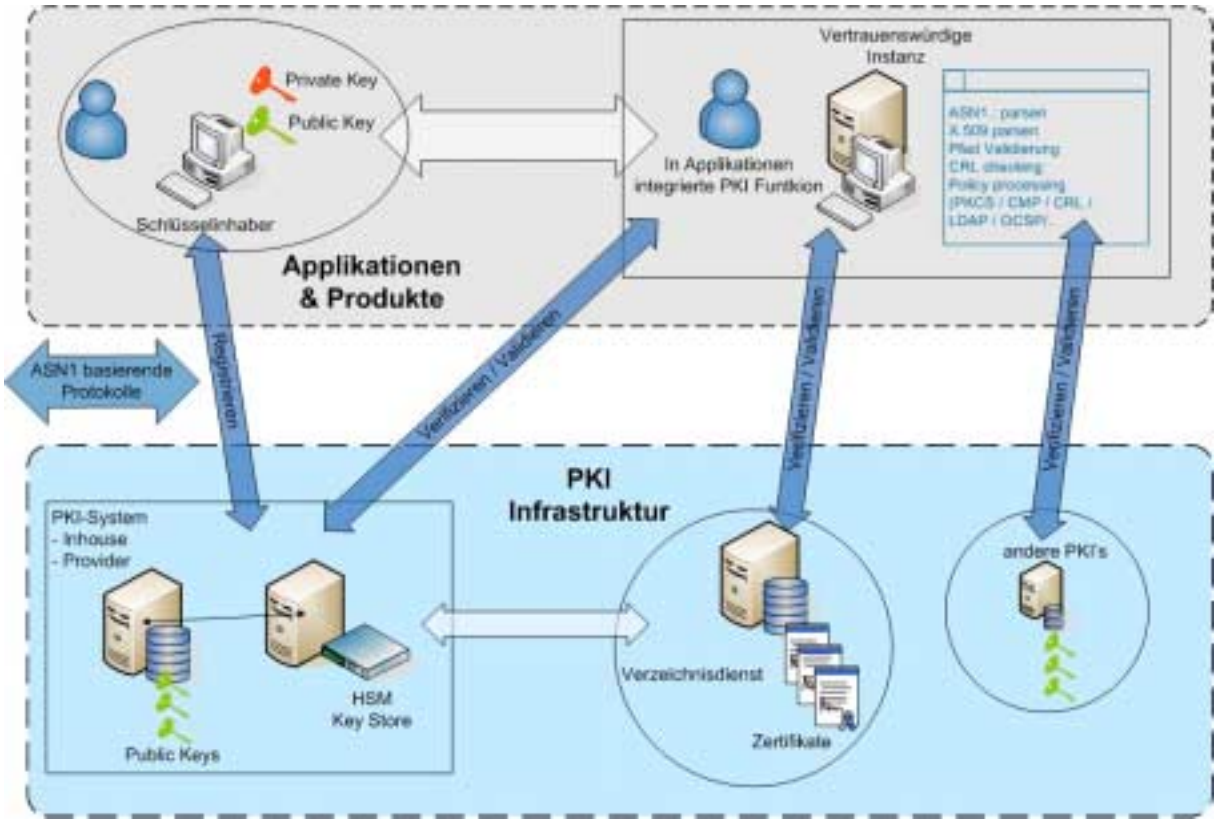

Abbildung 2: Übersicht einer PKI Infrastruktur

In der bisherigen PKI-Geschichte wurde der Schwerpunkt auf die Zertifikatsverwaltung gesetzt – nicht auf Software, die diese auch nutzt. Die Infrastruktur war vorhanden, doch existierten Anwendungen nur in geringem Umfang. Dieses Problem hat sich heute relativiert, jedoch fehlt nach wie vor die zwingende Anwendungen für eine PKI: es existiert keine "Killerapplikation" für die breite Masse. Zudem ist die Integration der PKI-Standards in Anwendungssoftware komplex und benötigt enormen Entwicklungsaufwand. Hier setzt das XKMS-Protokoll mit seinem Ansatz an, die Komplexität der Zertifikatsvalidierung und -verwaltung an eine zentrale Stelle zu deligieren. Basierend auf den Verfahren und Mechanismen von PKIX und Web Services Security besteht nun die Möglichkeit, dass Anwendungen im Rahmen von Geschäftsprozessen die Vorteile von zertifikatsbasierender Security nutzen und anwenden können.

3 XKMS Dienste

Grundgedanke der XKMS-Spezifikation ist es, eine erweiterte Menge von XML-Definitionen anzubieten, um ein vollständiges Interface zu gängigen PKIs intern oder auch zu Drittanbietern zu realisieren. Die nachfolgende Grafik zeigt in Anlehnung an das PKIX-Model (vergleiche Abbildung 2) den Ansatz von XKMS.

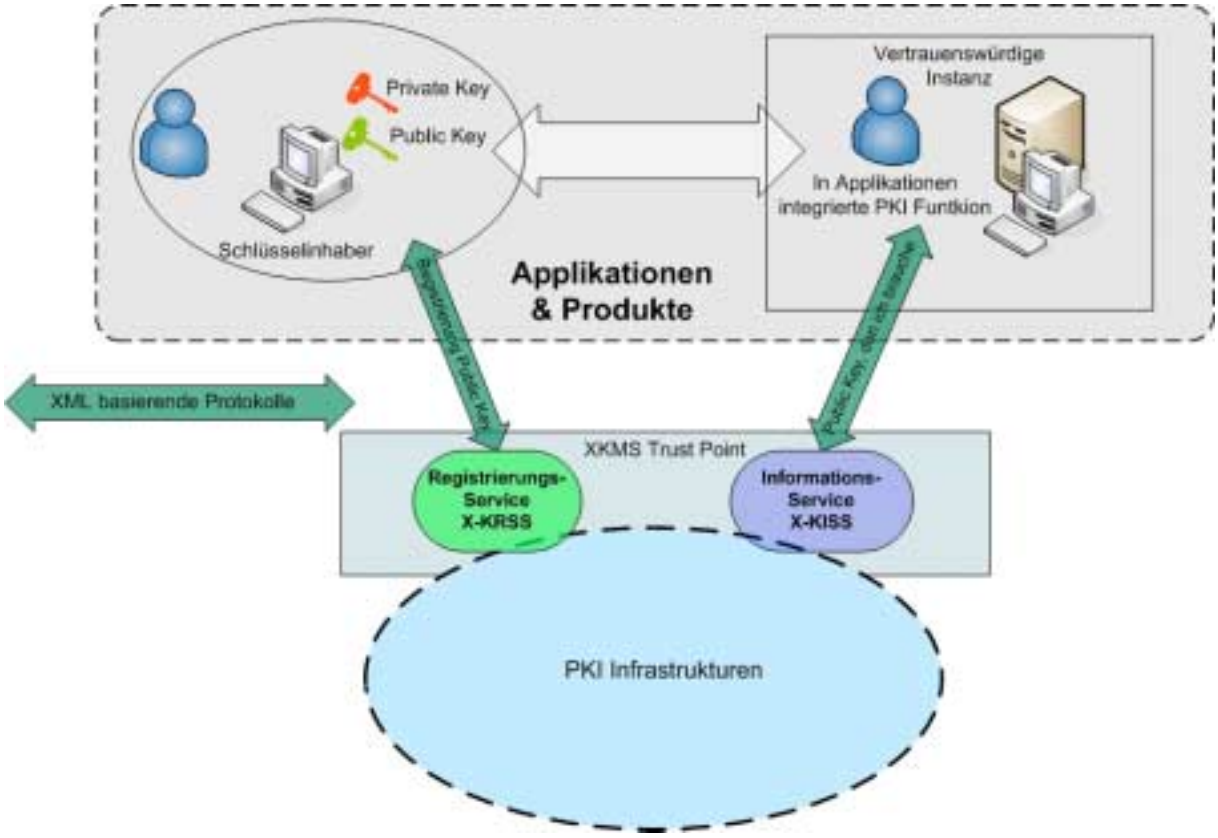

Abbildung 3: XKMS Modell

Mit XKMS werden grundsätzlich zwei Dienste zur Verfügung gestellt:

- *XML Key Registration Service Specification (X-KRSS)*, spezifiziert den Lebenszyklus von Schlüsseln (Registrierung, Widerruf, Neuauflage) und gegebenenfalls auch die Wiedergewinnung von zugehörigen privaten Schlüsseln.
- *XML Key Information Service Specification (X-KISS)*, spezifiziert die Anfrageoperationen in Bezug auf die Verifizierung von öffentlichen Schlüsseln und der zugehörigen Zertifikate.

Grundlage dieser Dienste ist das XKMS Protokoll, das ein *Request/Response*-Verfahren auf Basis von SOAP implementiert.

(Anmerkung: In der Version 2.0 ermöglicht XKMS so genannte *Compound Requests*, also eine Anfrage mit mehreren Operationen bzw. Zertifikaten durch einen Client, sowie die asynchrone Verarbeitung von Anfragen.)

Wie kann nun ein möglicher Anwendungsfall von XKMS aussehen? Ein Web Service oder eine Applikation generiert entsprechende X-KISS/XKRSS-Anfragen und leitet diese an eine

vertrauenswürdige Zwischeninstanz (im weiteren "TrustPoint" genannt) zur Verarbeitung weiter. Der TrustPoint, ebenfalls ein Web Service, bildet die Schnittstelle zu existierenden Public Key Infrastrukturen. Er interpretiert die von XKMS spezifizierten XML-Tags und kann so beispielsweise einen angeforderten öffentlichen Schlüssel lokalisieren und in Form eines Zertifikates zurückgeben oder das empfangene Zertifikat interpretieren und validieren. In der nachfolgenden Grafik wird dieses Verfahren verdeutlicht. Es wird hier schon entsprechend den Zielsetzungen in dem Projekt zur Realisierung des TrustPoints die European Bridge-CA mit integriert.

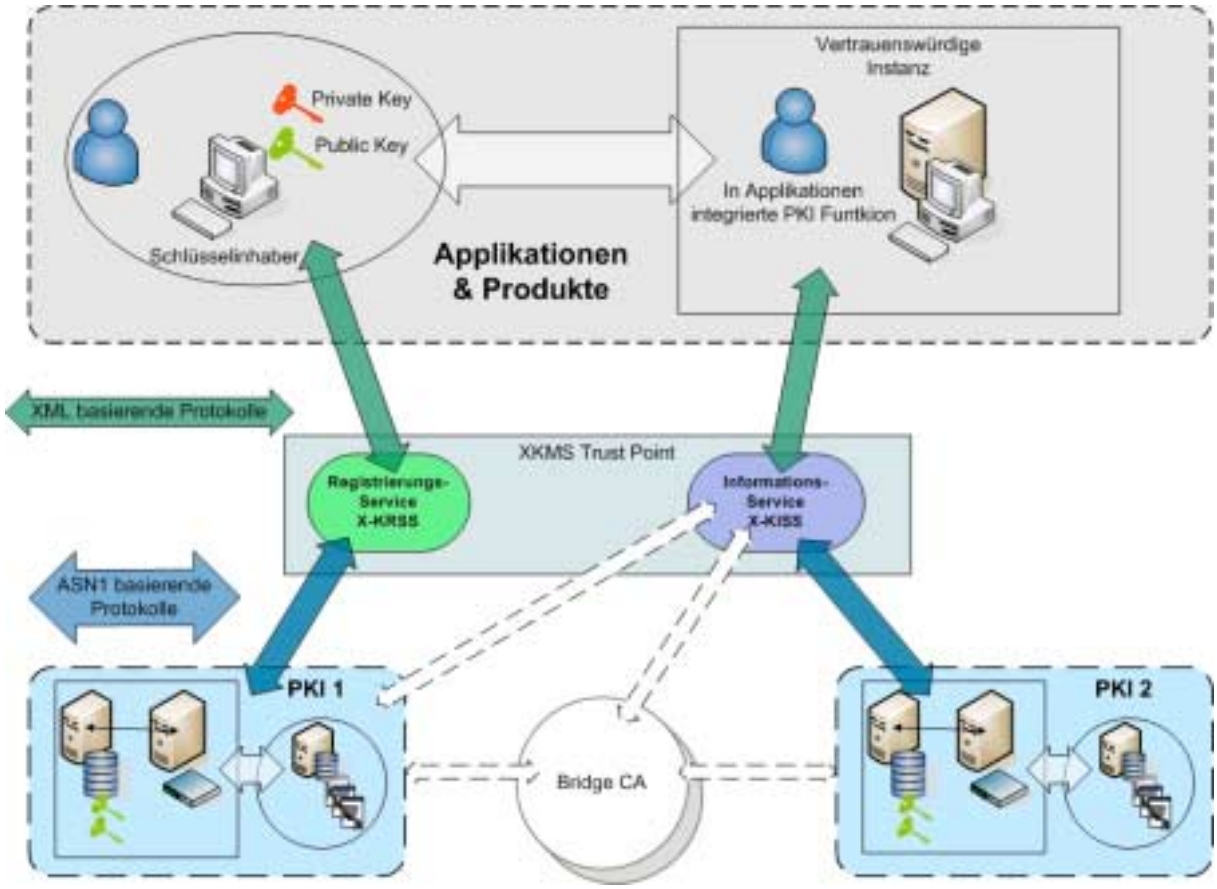

Abbildung 4: XKMS im Überblick

3.1 Vor- und Nachteile von XKMS-Diensten

In Rahmen dieses Kapitels wird eine Übersicht hinsichtlich der Vor- und Nachteile von XKMS-Diensten gegeben. Hierbei ist zu beachten, dass die aufgeführten Nachteile sich ausschließlich auf Probleme bzgl. der Absicherung von Web Services im Allgemeinen beziehen.

Vorteile :

- Reduzierte Komplexität auf Client-Seite
  Eine breite Palette von Anwendungen und Geräten kann so die Funktionen einer PKI nutzen. Beispielsweise können Pocket PCs oder Embedded Systems mit geringer Prozessorleistung und vergleichsweise wenig Arbeitsspeicher den kompletten PKI-Funktionsumfang nutzen.

- Einfachere Implementierungen
  Implementierungen für verschiedene Betriebssysteme fallen weg. Die Funktionalität wird zentral bereitgestellt.
- Einfaches Roll-Out
  Da die eigentliche Arbeit serverseitig geschieht, sind neue Funktionen "on the fly" für alle Nutzer ohne clientseitige Änderungen verfügbar.
- Zentrales Trust Management
  Es kann eine zentrale Trust-Policy definiert und einfach umgesetzt werden.
- Offene Standards
  Die XKMS Spezifikation setzt auf offene XML-Standards, die von jedem Betriebssystem und Programmiersprache genutzt werden können.
- Zukunftssicherheit
  XKMS-Dienste werden zentral durch offene Standards bereitgestellt. Neue Entwicklungen können so schnell und zentral implementiert werden. Anwendungen müssen nicht oder nur sehr gering angepasst werden.

Nachteile:

- DOS-Attacken

  XKMS Dienst sind wie alle Web Services, sollten Sie frei verfügbar sein, anfällig gegen DOS Attacken. Abhilfe schaffen hier Web Service Firewalls und SOAP Firewalls.
- Single Point of Failure

  Durch die Zentralisierung des Trust-Services steht gerade die Verfügbarkeit und Ausfallsicherheit im Vordergrund. Hier sind bei Design und Betrieb des Systems auf jeden Fall schon in der Konzeptionsphase entsprechende Vorkehrungen zu treffen.

4 Projekt TrustPoint

Ziel des Projektes TrustPoint ist der Aufbau und Betrieb eines XKMS Responders. Der Aufbau des TrustPoints wird komplett durch Open Source Komponenten vorgenommen. Freie Verfügbarkeit und Einsicht in Quelltexte sind wichtige Eigenschaften bei der Entwicklung sicherheitsrelevanter Software.

Das Projekt TrustPoint gliedert sich, nach dem Stand der derzeitigen Planung in zwei Projekt-Phasen.

Im Folgenden wird kurz die Zielsetzung innerhalb der einzelnen Phasen beschrieben:

Phase I:

Im ersten Schritt werden die Architektur und das Systemdesign des XKMS Responders definiert. Neben der Auswahl der Hard- und Softwarekomponenten, stehen ebenfalls die anzubindenden Infrastrukturen im Vordergrund. Ebenso im Vordergrund steht schon in der ersten Planungsphase die Integration der European Bridge-CA (EBCA). Der zu realisierende Funktionsumfang des XKMS Responders in der Projektphase I beschränkt sich auf die Funktionen locate und validate von X-KISS.

Phase II:

Im Rahmen der Projektphase II wird der Funktionsumfang des TrustPoints um die Funktionen von X-KRSS erweitert. Zudem werden betriebstechnisch die Möglichkeiten des Outsourcings betrachtet.

4.1 TrustPoint Phase I: Architektur und Spezifikation

Projektphase I beschränkt sich auf die X-KISS-Funktionen locate und validate. Im ersten Schritt wird die Übertragung der SOAP-Nachrichten und Auswertung der XKMS-Funktionen realisiert. Die Zertifikate werden dann zur Überprüfung an eine lokale PKI geleitet. Durch die Anbindung des TrustPoints an die European Bridge-CA können die zentralen Bridge-Verzeichnisdienste und Bridge-Überprüfungsdienste genutzt werden.

Die Architektur des XKMS Responders gliedert sich in vier logische Module:

Abbildung 5: Module des TrustPoints

Transport-Protokoll:

Im Rahmen dieses Moduls werden die notwendigen Transportprotokolle zur Kommunikation mit dem XKMS Responder bereitgestellt. In der Projektphase I wird ausschließlich HTTP bzw. HTTPS realisiert.

Zur Nachrichtenübertragung über das HTTP-Protokoll wird Apache Tomcat eingesetzt. Tomcat ermöglicht durch verschiedene so genannte Connectors unter anderem die Verbindung durch HTTP und HTTPS.

SOAP Verarbeitung:

Im Rahmen der SOAP-Verarbeitung werden basierend auf SOAP V.1.2 alle Funktionen bereitgestellt, um die eingebetteten XML Nachrichten weiter zu verarbeiten.

Zur Verarbeitung der SOAP-Anfragen und Antworten wird Apache AXIS eingesetzt. SOAP bettet die XML-Nachrichten in das SOAP-Body Element innerhalb des SOAP-Envelopes ein. Der optionale SOAP-Header wird dabei nicht benutzt. SOAP stellt mit SOAP-Faults geeignete Fehlermeldungen und -beschreibungen zur Verfügung. Mit Apache AXIS steht eine ausgereifte Implementierung des SOAP-Protokolls V.1.2 zur Verfügung. AXIS bietet ein Framework für die Entwicklung von Clients und Servern und ist unabhängig vom verwendeten Transportprotokoll. Im Projekt TrustPoint läuft AXIS als Servlet innerhalb des Servlet-Containers von Apache Tomcat, wobei AXIS das ent- und verpacken der SOAP Nachrichten übernimmt und für jeden Web Service eine WDSL bereitstellt.

XML Verarbeitung

Zur weiteren Verarbeitung der XML-Nachrichten werden durch entsprechend realisierte Instanzen der X-KISS-Klassen die XKMS Funktion extrahiert. Ebenfalls wird über die X-KISS-Klassen die notwendige Funktion zur Erzeugung der Antwortnachricht bereit gestellt. Um optional Nachrichten zu signieren, wird auf die XML Security API des Apache XML-Projektes zurückgegriffen. Dies ermöglicht dann das optionale Signieren der XKMS-Antworten.

PKI-Plug-In

Durch die unterschiedlichen PKI-Plug-Ins werden dann die in den X-KISS Nachrichten enthaltenen Zertifikate zur Überprüfung weitergeleitet. Hierzu werden die gemäß PKIX spezifizierten herkömmlichen PKI-Protokolle und Methoden verwendet.

Die folgende Grafik stellt zusammenfassend eine Übersicht über die Software Architektur des TrustPoints dar wie er in Phase I realisiert ist.

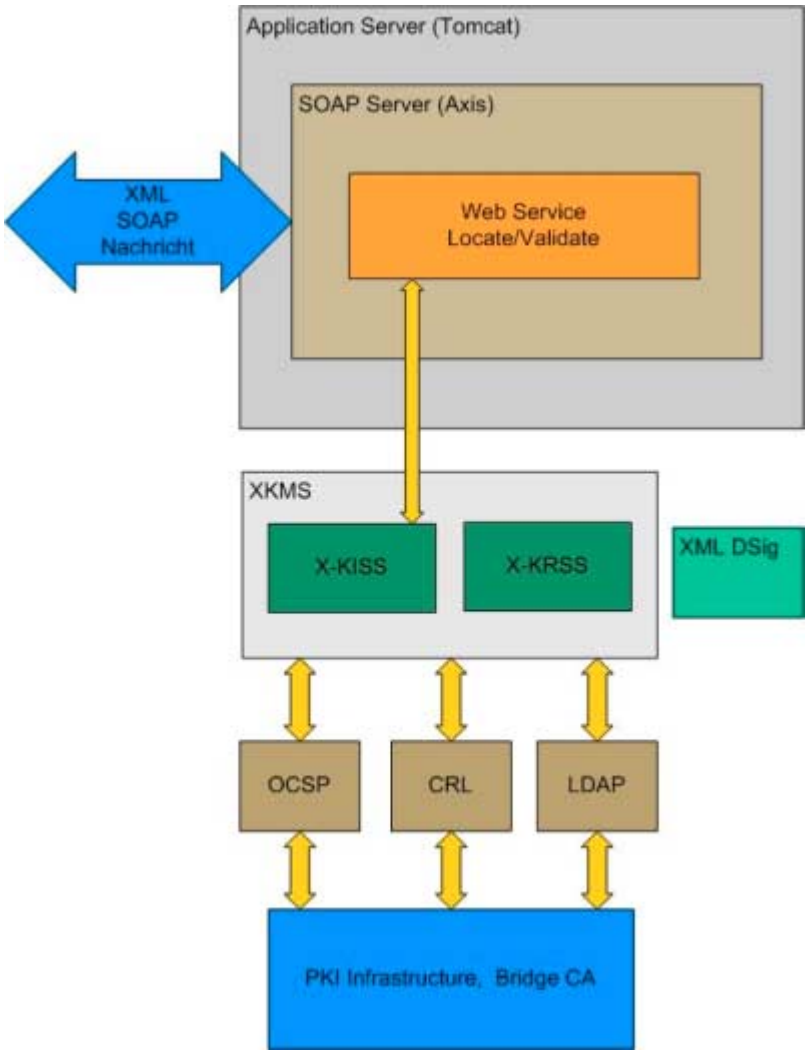

Abbildung 5: Architektur der Software des TrustPoints Phase I

4.2 TrustPoint Phase II: Erweiterung der Funktionen

Nach Abschluss der Phase I steht nun ein funktionierender XKMS Responder zur Verfügung. Zertifikate können durch den TrustPoint gefunden und validiert werden.

In Phase II des Projektes soll der TrustPoint um die Funktionalitäten der X-KRSS Spezifikation erweitert werden. X-KRSS definiert Funktionen um Schlüssel zu registrieren, neu auszustellen und zurückzuziehen. Dabei soll es möglich sein, dass der Client oder der TrustPoint ein Schlüsselpaar erzeugt. Wird der Schlüssel von dem TrustPoint erzeugt, muss der private Schlüssel sicher übertragen werden was durch die Verwendung von XML- Encryption sichergestellt wird.

Neben der funktionalen Erweiterung des TrustPoints, steht im Rahmen der Projekt Phase II die Integration und Umsetzung von unterschiedlichen Policies im Vordergrund. Hierzu sollten im ersten Schritt Modelle und Möglichkeiten evaluiert werden, die es erlauben standardisiert unterschiedliche Vorgaben, Mechanismenstärken und auch Verfahren zu bündeln.

5 Ausblicke

Ausblickend auf die Erwartungen und Einsatzgebiete von zentralen Services zur Sicherung von elektronischen Geschäftsprozessen, ist zunächst einmal zu berücksichtigen, da Stand der Ausarbeitung die XKMS Spezifikation sich noch im Status Candidate Recommendation befindet, und die Release der Spezifikation Ende 2004 erwartet wird.

Unterschiedlichste Anwendungen und Einsatzgebiete werden heute schon diskutiert, wobei hinsichtlich des Einsatzes von XKMS im Bezug auf mobile Devices immer die zunehmende Leistungsstärke der nächsten Generationen der Endgeräte zu beachten ist.

Mit der Umstellung und Einführung von Web Services innerhalb von ERP und DMS Systemen werden heute die Weichen gestellt um Geschäftsprozesse effizienter und modularer zu gestalten. Die Vertrauenswürdigkeit beim Einsatz von Web Services kann durch den Einsatz von TrustPoints sichergestellt werden, die alle notwendigen Funktionen ebenfalls als Web Service zu Verfügung stellen. Somit ergibt in der Zukunft, basierend auf der zu Beginn gezeigten Darstellung (Abbildung 1) folgendes Szenario.

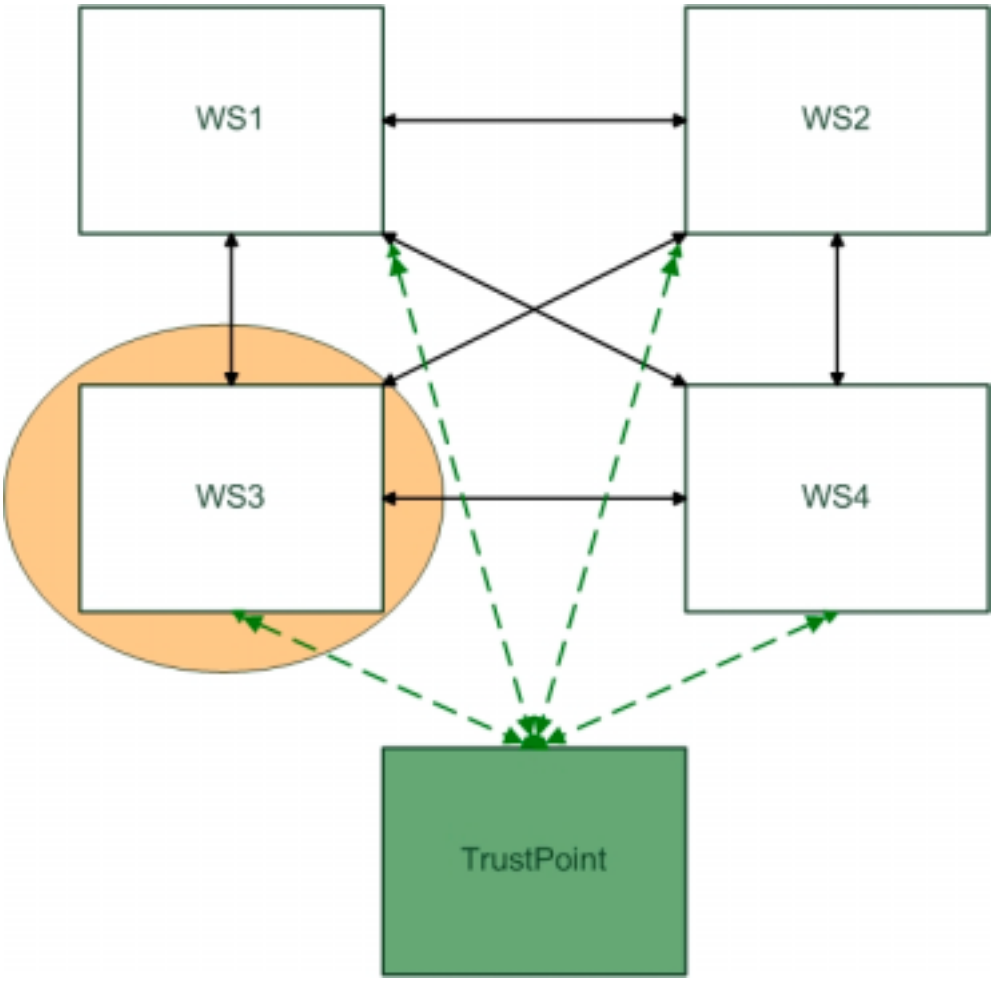

Abbildung 6: Web Services Szenario mit TrustPoint

Literatur

[W3C04] W3C: XML Key Management Specification (XKMS 2.0), Candidate Recommendation 5. April 2004, http://www.w3.org/TR/xkms2/

[Neil03] M. O` Neill: Web Services Security, Osborne Verlag (2003)

[Nash02] A. Nash, W. Duane, C. Joseph, D. Brink: PKI E-security implementieren, mitp Verlag (2002)

[Snell02] J.Snell, D. Tidwell, P.Kulchenko: Webservice-Programmierung mit SOAP, O`Reilly Verlag (2002)