Institut für Informatik
Professur ABVS / Telematik
Johann Wolfgang Goethe-Universität
Frankfurt am Main

Fraunhofer IPSI
Institut
Integrierte Publikations-
und Informationssysteme

Diplomarbeit

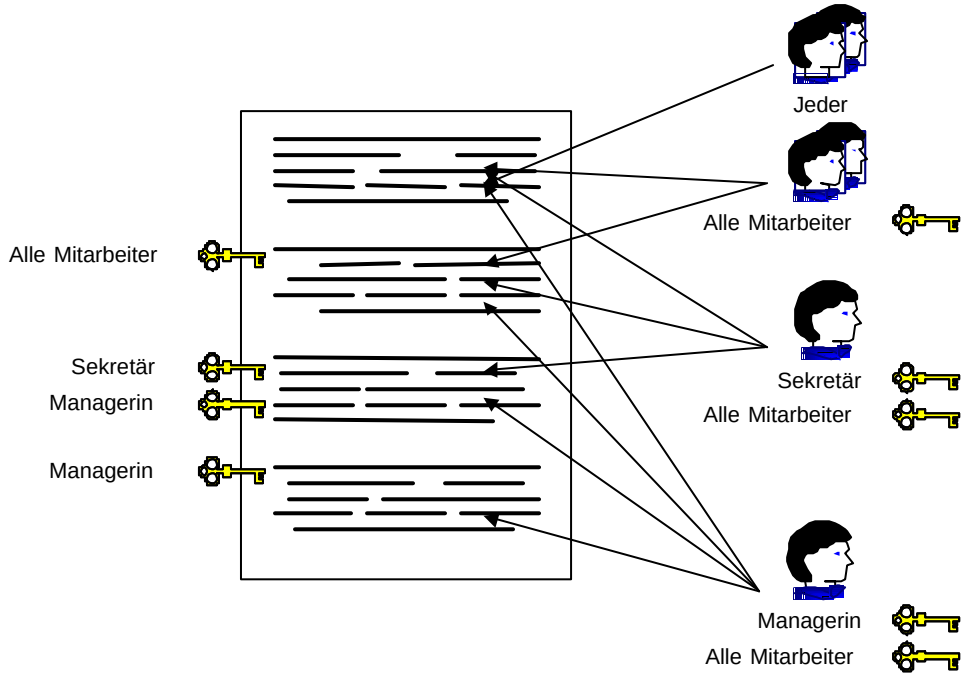
Entwurf und prototypische Realisierung einer Architektur zur flexiblen Verschlüsselung von XML-Daten

Arne Priewe

IPSI: Gerald Huck

Motivation und Ziele

- Vertraulichkeit durch Verschlüsselung
  - feingranular
  - hierarchisch
  - verarbeitbar
- Architektur zur Verschlüsselung
  - flexibel
  - deklarativ
  - verteilt pflegbar
  - erweiterbar



Inhalt

- Verschlüsselung von XML-Daten
- Generische Architektur zur Verarbeitung von XML-Daten
- Architektur zur Verschlüsselung von XML-Daten
- Prototypische Realisierung
- Demonstration
- Ausblick
- Diskussion

Verschlüsselung von XML-Daten

Repräsentation und Granularität

- Repräsentation verschlüsselter XML-Daten wiederum in XML
  - Super-Verschlüsselung (Verschlüsselung von verschlüsselten XML-Daten)
  - Hierarchische Verschlüsselung
- Granularität
  - Element `<root>text<child attr="value" /></root>`
  - Attribut `<root>text<child attr="value" /></root>`
  - Text `<root>text<child attr="value" /></root>`

Platzierung verschlüsselter Inhalte

- Prinzip der Stabilität:
  für Verarbeitbarkeit essenziell wichtig
- Verschlüsselte Inhalte ersetzen die Daten direkt am vorherigen Ort
- Positionierung verschlüsselter Attribute „schwierig“, weil nie völlig stabil:
  Wird als direktes Kind-Element platziert.

Serialisierung und Deserialiserung

- Vollständige Unterstützung von XML-Namensräumen
- Elemente werden direkt nach XML serialisiert
- Attribute benötigen ein spezielles Serialisierungsschema

```
<element ns:attr="wert" xmlns:ns="http://cs.uni-frankfurt.de/meinNS">
         ns:attr="wert" http://cs.uni-frankfurt.de/meinNS
```

- Text wird direkt serialisiert
- Zeichenkodierung erfolgt immer mit UTF-8

Verschlüsselung (1)

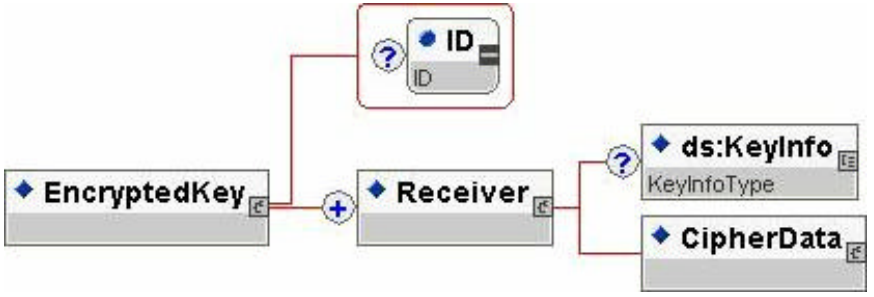
- Verschlüsselungs-Schemas nach CMS-Spezifikation (RFC 2630 – Cryptographic Message Syntax)
- DES-ede mit CBC-Modus
  - 168 Bits, 8 Oktets Initialisierungsvektor (IV)
  - Padding nach PKCS #5 (RFC 2898)
- RSAES-PKCS1-v1_5
  (RFC 2437 – PKCS #1: RSA Cryptography Specifications Version 2.0)
  - mind. 1024 Bits
- Spezifikation der Verfahren mittels URLs

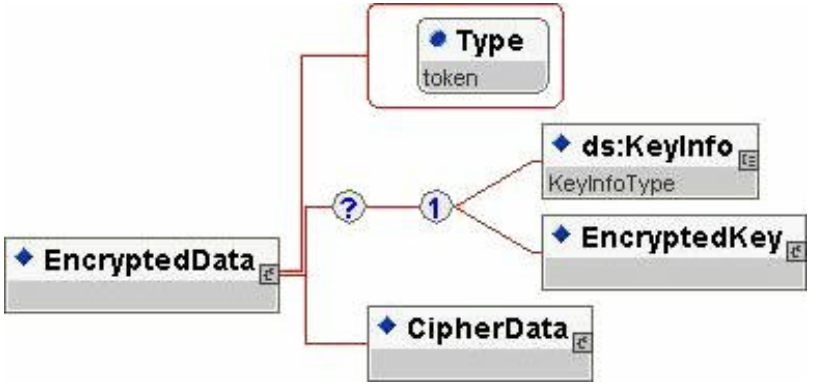
Verschlüsselung (2)

- XML-Daten mit symmetrischen Verfahren
  - Geteilter Schlüssel oder Sitzungsschlüssel
- Sitzungsschlüssel mit asymmetrischen Verfahren
- Schlüsseltransport für Sitzungsschlüssel
  - innerhalb des Kommunikationsbandes
  - ausserhalb des Kommunikationsbandes
- Wiederverwendung von Sitzungsschlüsseln

Schema (Übersicht)

- EncryptedKey
  - Verschlüsselte(r) Sitzungsschlüssel
  - Meta-Daten zur Entschlüsselung
  - KeyInfo wird von XML-Signature (W3C) importiert
- EncryptedData
  - Verschlüsselte XML-Daten
  - Meta-Daten zur Entschlüsselung



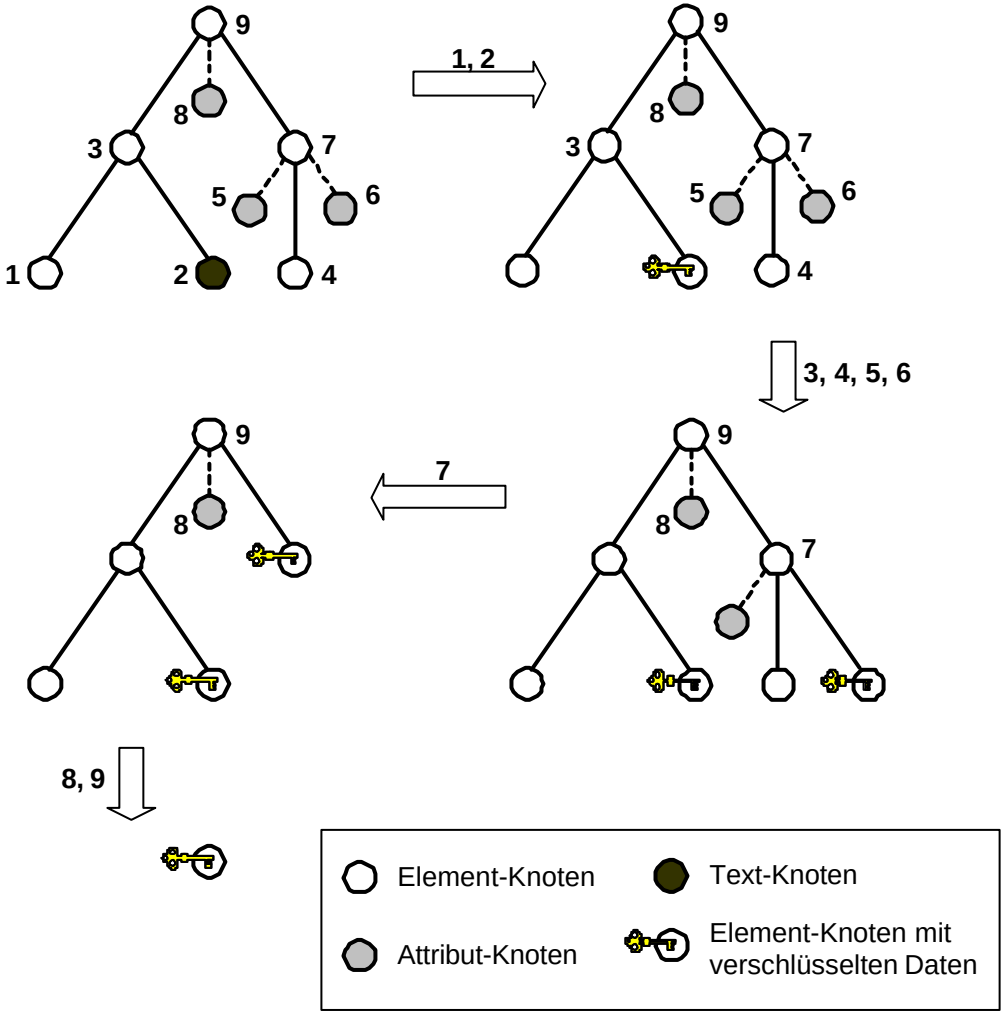


Verschlüsseln von XML-Daten

- Randbedingungen berücksichtigen
  - Keine Entitäten-Referenzen
  - Verarbeitungsvorschriften (bitte warten...)
- Super-Verschlüsselung
  - Implizite Festlegung eine Reihenfolge für die Entschlüsselung
- Hierarchische Verschlüsselung
  - Bei Elementen möglich
  - Definition basiert auf dem DOM (W3C)

Hierarchische Verschlüsselung

- Attribute erfordern eine Erweiterte Postorder-Traversierung
- Beispiel für die Verschlüsselung der Knoten 2, 6, 7, und 9



Entschlüsseln von XML-Daten

- Randbedingungen berücksichtigen
  - Keine Entitäten-Referenzen
  - Inverse Transformationen (siehe Arbeit)
  - Verarbeitungsvorschriften (bitte warten...)
- Entschlüsselte XML-Daten in das Dokument einfügen
  - Zeichenkodierung von UTF-8 in die des Dokuments konvertieren (falls notwendig)
  - Bei Attributen vorhandene XML-Namensräume und Präfixe berücksichtigen und bei Bedarf anpassen

Verarbeitungsvorschriften

- Einfüge-, Lösch- und Änderungsoperationen
- Atomizität verschlüsselter Daten
- Wahrung der referenziellen Integrität
  - Verschlüsselte Attribute
  - Referenzen von KeyInfo-Elementen auf EncryptedKey-Elemente im gleichen Dokument
- Randbedingungen
  - Doppelte Attribute
  - Eineindeutigkeit von Referenzen

XML Encryption (W3C) im Vergleich

- Teilweise noch in der Entwurfsphase
- Granularität (binäre Datenobjekte, XML-Dokumente, Elemente, Element-Inhalte)
- Reichhaltigere Auswahl bei kryptografischen Verfahren
- Mehrere Empfänger nur mittels geteilter Schlüssel
- Hierarchische Verschlüsselung nicht explizit spezifiziert
- Keine expliziten Verarbeitungsvorschriften spezifiziert
- Kein Verfahren zur Serialisierung und Deserialisierung spezifiziert

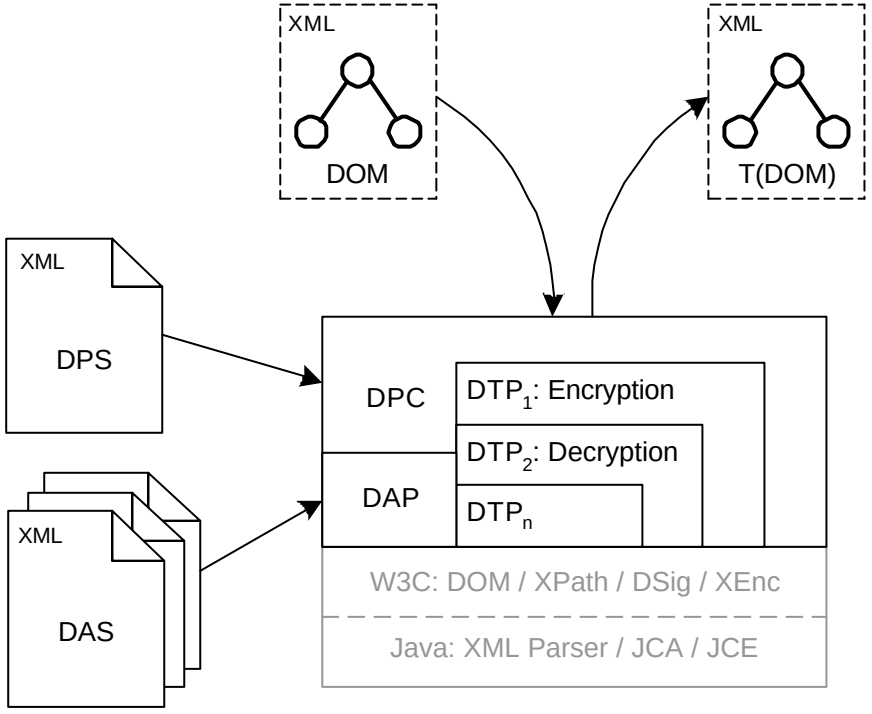
Generische Architektur zur Verarbeitung von XML-Daten

Überblick (1)

- Verwendet ausschliesslich XML / XML-Technologien
- Transformation von XML-Dokumenten mittels Transformations-Prozessoren (auf DOM basierend)
- Meta-Daten (*Annotationen* genannt) können die Transformations-Prozessoren steuern
- Annotations-Prozessor als generische Komponente zur Anbringung von Annotationen
- XPath als Selektions-Sprache für Annotationen
- Konfigurierbare Steuerungs-Komponente zur Kontrolle des Datenflusses (Prinzip der *Kette von Werkzeugen*)

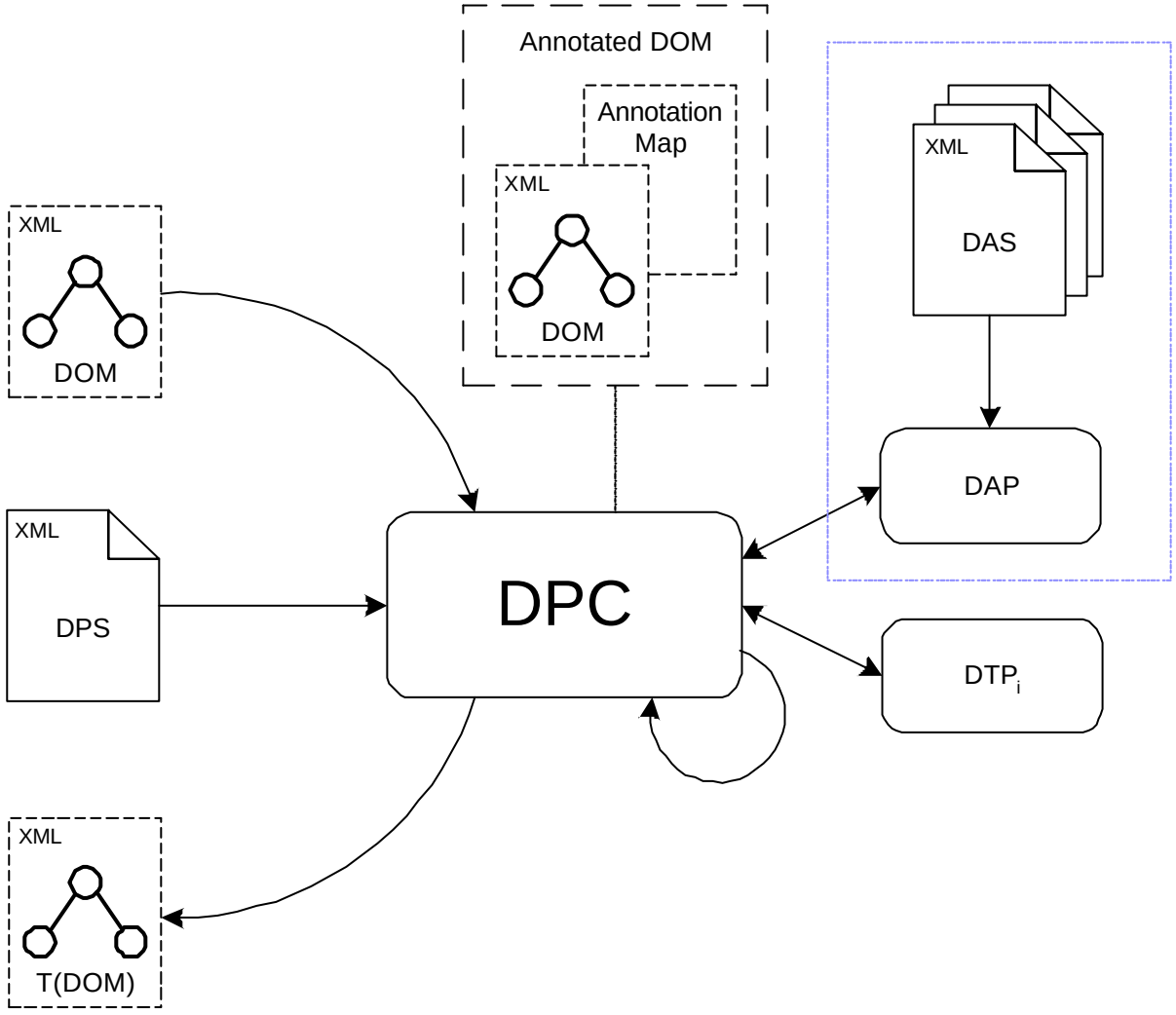
Überblick (2)

Document Processing Framework (DPF)



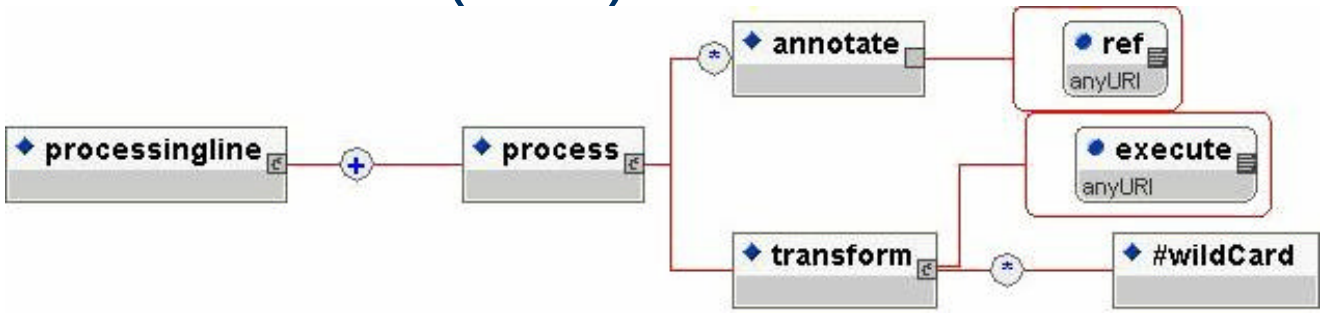
Document Processing Controller (DPC)

- Steuerungs-Komponente
- Konfiguration über ein DPS



Document Processing Sheet (DPS)

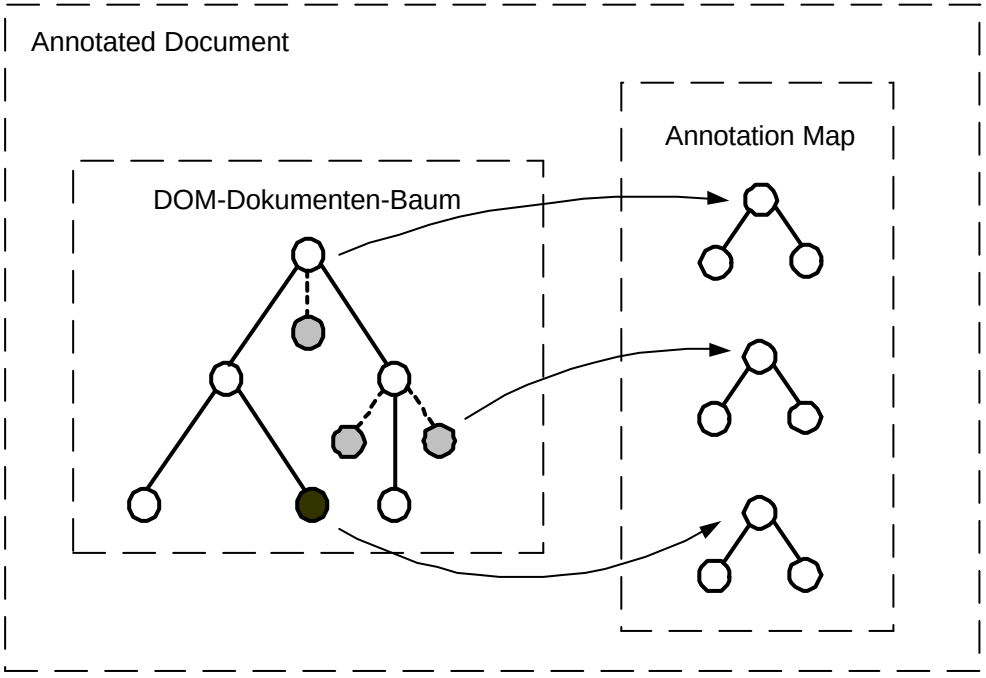
- Deklarative Konfiguration des DPCs mittels eines XML-Dokuments (DPS)

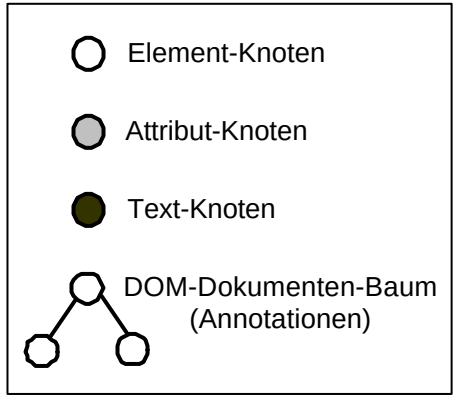


- Mehrere Transformations-Prozessoren (DTPs) können hintereinander ausgeführt werden (Prinzip der Kette von Werkzeugen)
- DTPs werden über URIs referenziert
- Weitere Argumente können an DTPs übergeben werden

Annotierter DOM-Baum

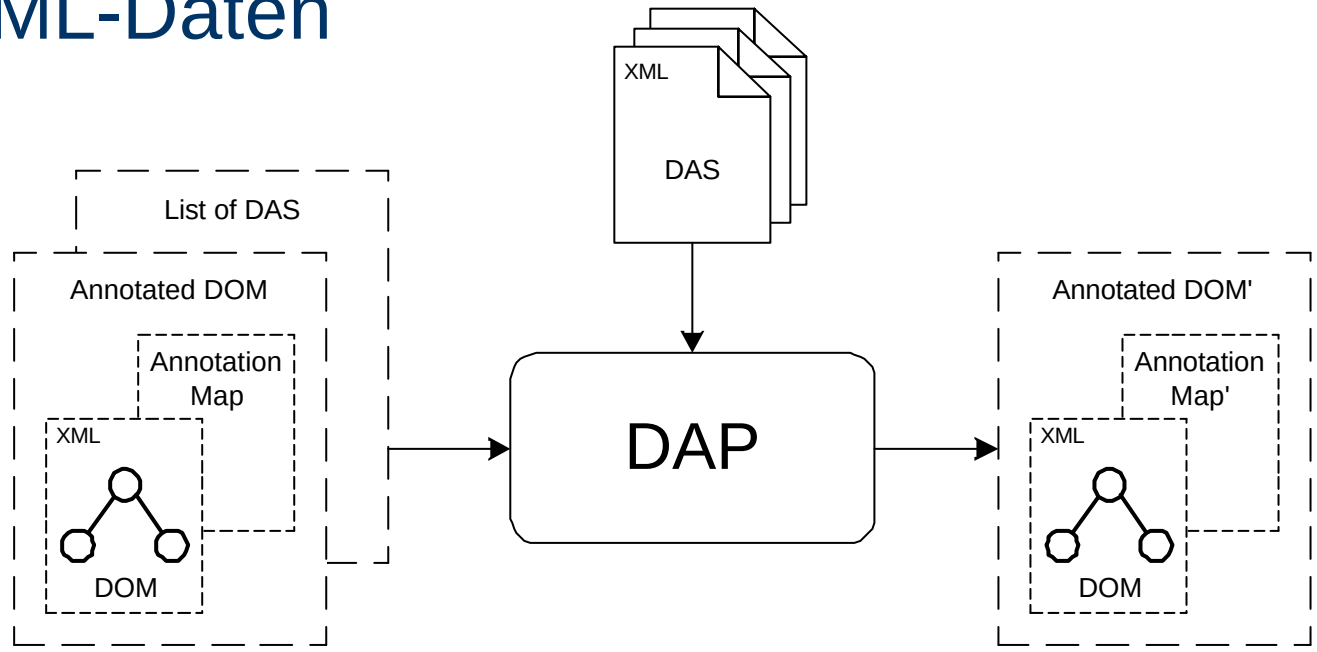
- Aggregation des DOM-Dokumenten-Baums und der zugeordneten Annotationen
- Alle Prozessoren setzen darauf auf





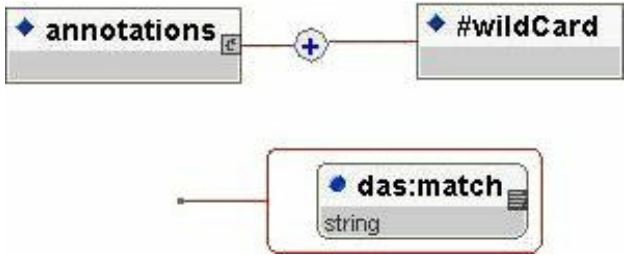
Document Annotation Processor (DAP)

- Fügt ausschliesslich Annotationen an die Knoten des annotierten DOM-Baum hinzu
- Ändert nie die im DOM repräsentierten XML-Daten



Document Annotation Sheet (DAS)

- Deklarative Spezifikation von Annotationen mittels eines XML-Dokuments (DAS)

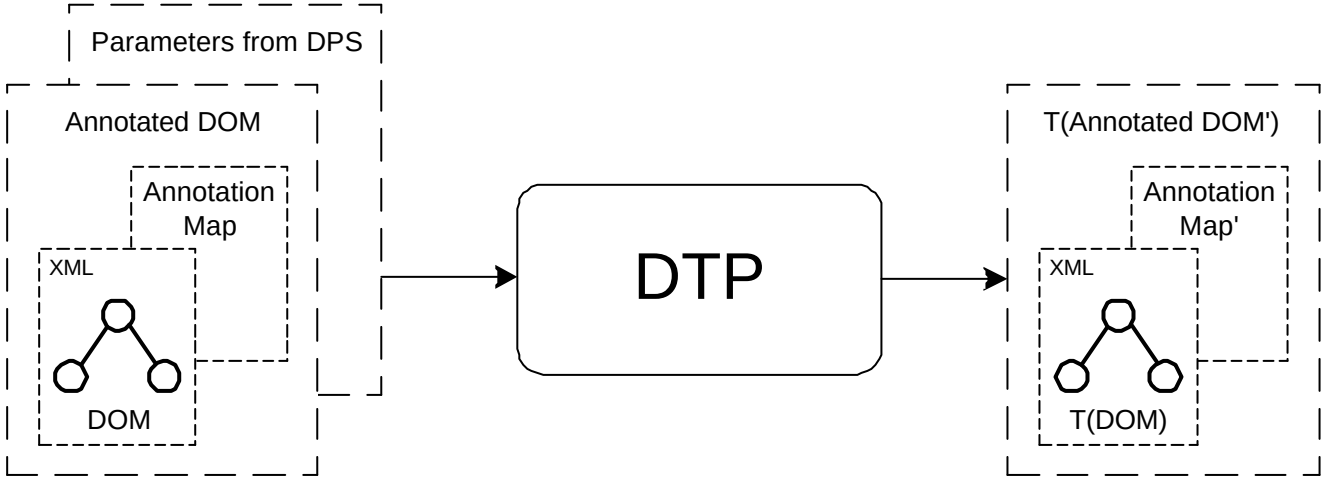


- Verteilte Pflege/Erstellung von Annotationen durch Verwendung mehrerer DAS-Dokumente
- Beinhaltet Annotationsregeln:

Annotation	(beliebige XML-Daten)
+	
Selektionsregel	(XPath-Ausdruck)

Document Transformation Processor (DTP)

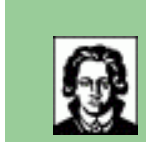
- Transformiert den annotierten DOM-Baum nach Belieben (also auch Annotationen!)
- Berücksichtigt dabei übergebene Parameter: Argumente + Annotationen



Erweiterbarkeit

- Durch Spezifikation / Implementierung von DTPs
- Jeder DTP spezifiziert
  - einen URI, der ihn eineindeutig identifiziert
  - die benötigten / möglichen Aufrufparameter:
    - Argumente für das DPS
    - Annotationen (z.B. per XML-Schema) - diese sollten immer einem eineindeutigen XML-Namensraum zugeordnet werden!
- Bestehende Transformations-Prozessoren können relativ leicht integriert werden, wenn diese auf dem DOM aufsetzen

Architektur zur Verschlüsselung von XML-Daten

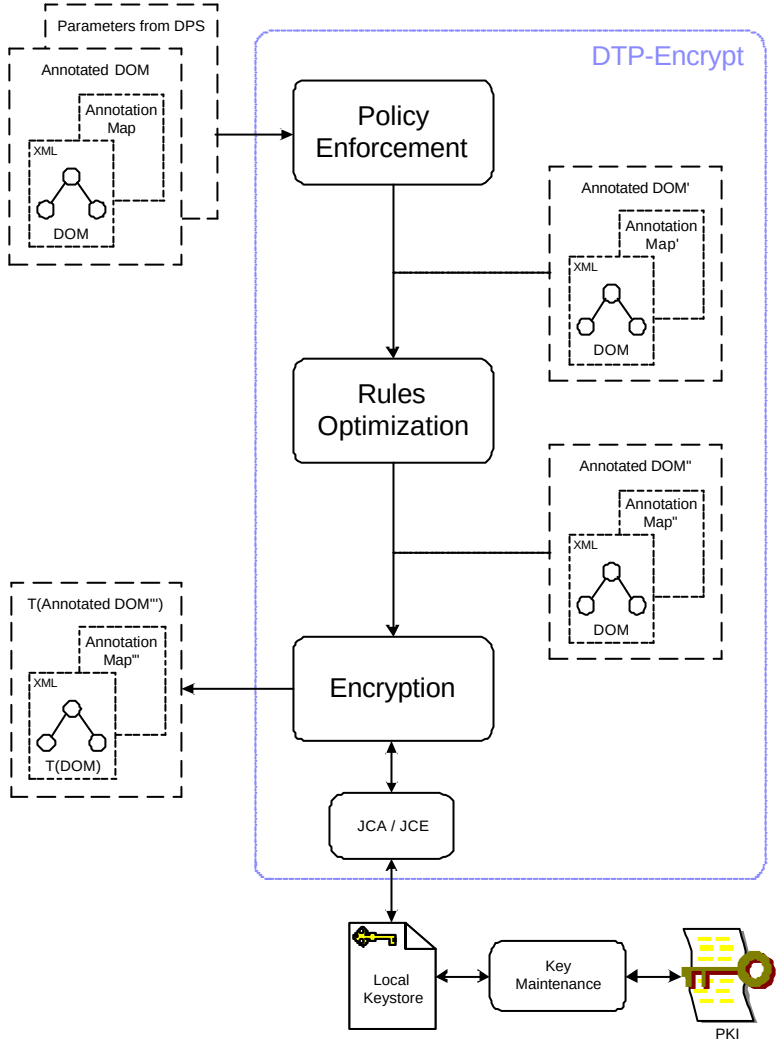


Überblick

- Verschlüsselung anhand deklarativer Regeln
- Sicherheitsrichtlinien (security policies) müssen berücksichtigt werden können
  - implizite Regeln (d.h. implizit erkennbar)
  - explizite Regeln (d.h. explizit angegeben)
- Optimierung von Verschlüsselungs-Regeln
- Unabhängig von der hier vorgestellten Spezifikation zur Verschlüsselung
- Baut auf dem DPF auf:
  DTPs zur Ver- und Entschlüsselung

DTP-Encrypt

- Besteht aus mehreren Unter-Komponenten
- Nur beim Verschlüsseln wird der DOM-Baum transformiert
- Die anderen Unter-Komponenten arbeiten nur auf den Annotationen

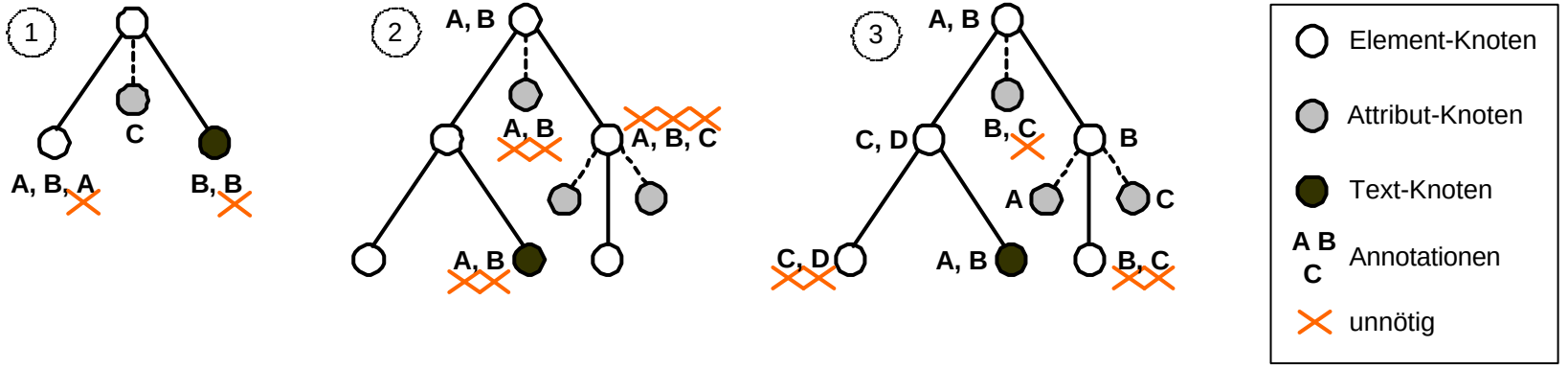


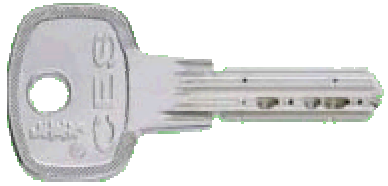
Policy Enforcement

- Durchsetzung von Sicherheits-Richtlinien anhand impliziter und/oder expliziter Regeln (z.B. als Annotationen)
- Ist in hohem Masse von der Einsatz-Umgebung abhängig, daher schwer generalisierbar
- Kann zur Garantie des Zugriffs auf verschlüsselte Daten eingesetzt werden (Vier-Augen-Prinzip)

Rules Optimization

- Entfernung unnötiger Verschlüsselungs-Regeln
  1. Redundante Annotationen zu einem Knoten des DOM-Baums: können immer entfernt werden
  2. Die Kinder eines Elements enthalten unnötige Annotationen: abhängig von der Anwendung (z.B. bei Workflow-Prozessen)



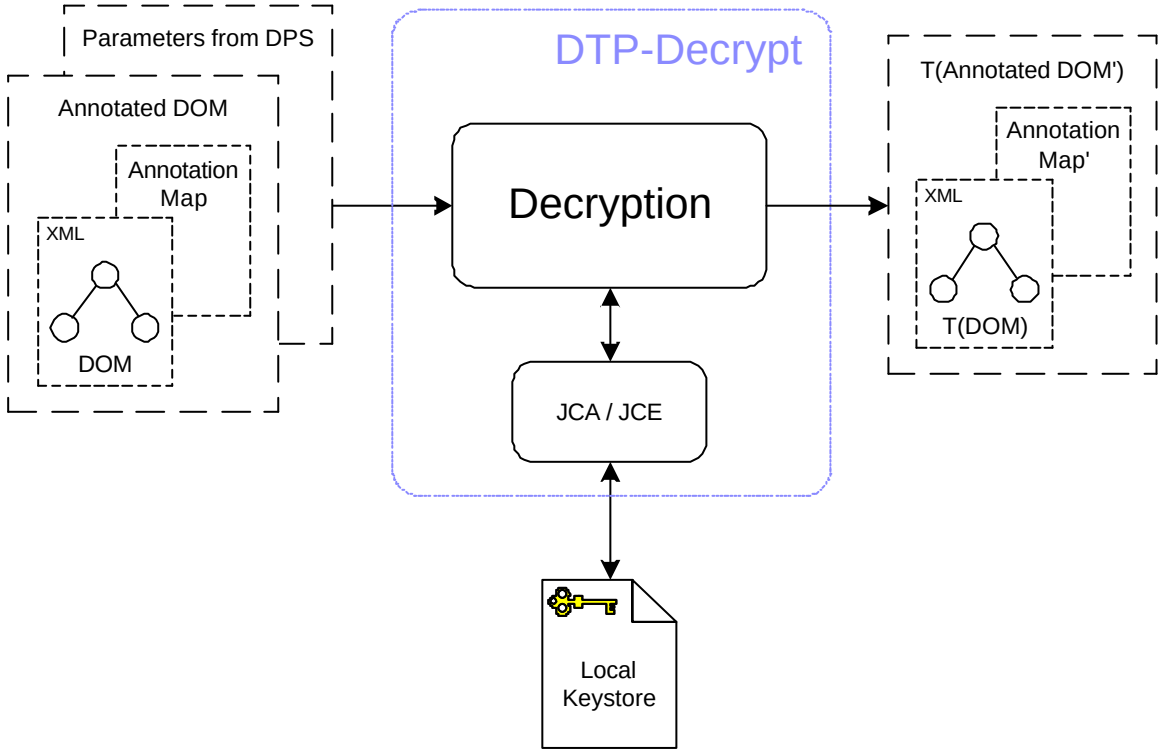


Encryption

- Verschlüsselt Knoten im DOM-Baum anhand von Annotationen und/oder Argumenten
- Erzeugt verschlüsselte Daten nach einer gegebenen Spezifikation (z.B. vom W3C)
- Garantiert ein nach den gegebenen Verschlüsselungs-Regeln adäquates Ergebnis (insbes. die Entschlüsselbarkeit)

DTP-Decrypt

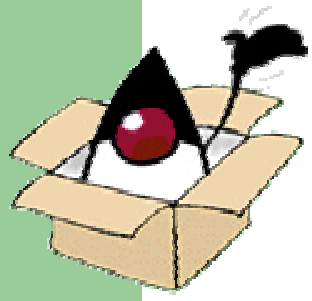
- Entschlüsselt verschlüsselte Daten anhand einer gegebenen Spezifikation



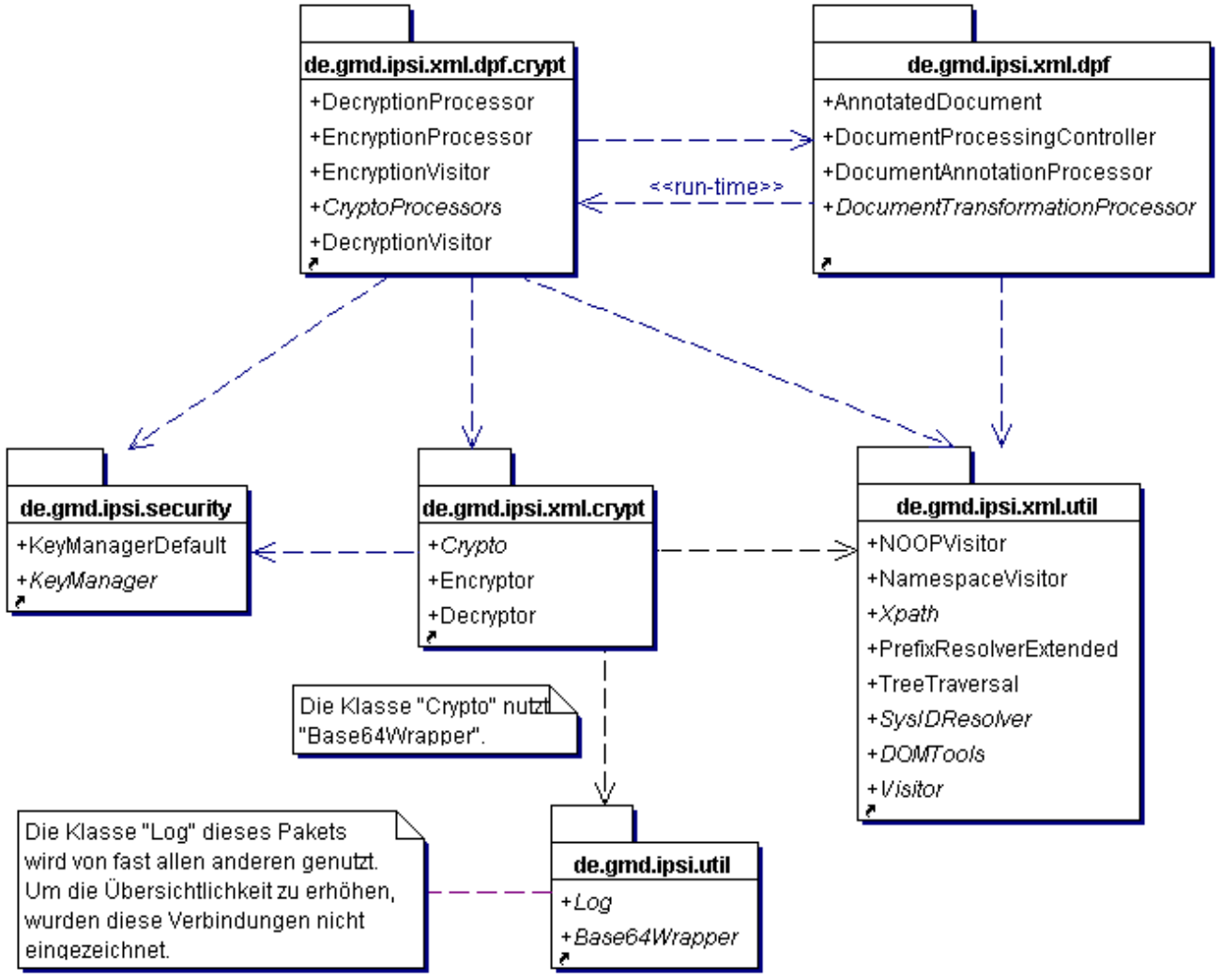
Prototypische Realisierung

Überblick

- Implementierung in Java
  - Java 2 SE v1.3 (Win2000)
  - Xerces 1.4.0 (XML-Parser)
  - Xalan 2.1.0 (XPath-Implementierung)
  - GMD/IPSI XML-Utilities (Serialisierung)
  - JCE-Implementierungen und Security-Provider
    - Bouncycastle Crypto Package 1.0.5
    - OpenJCE 1.1
    - Cryptix JCE v20000905-snap
    - Baltimore KeyTools Lite v5 for Java

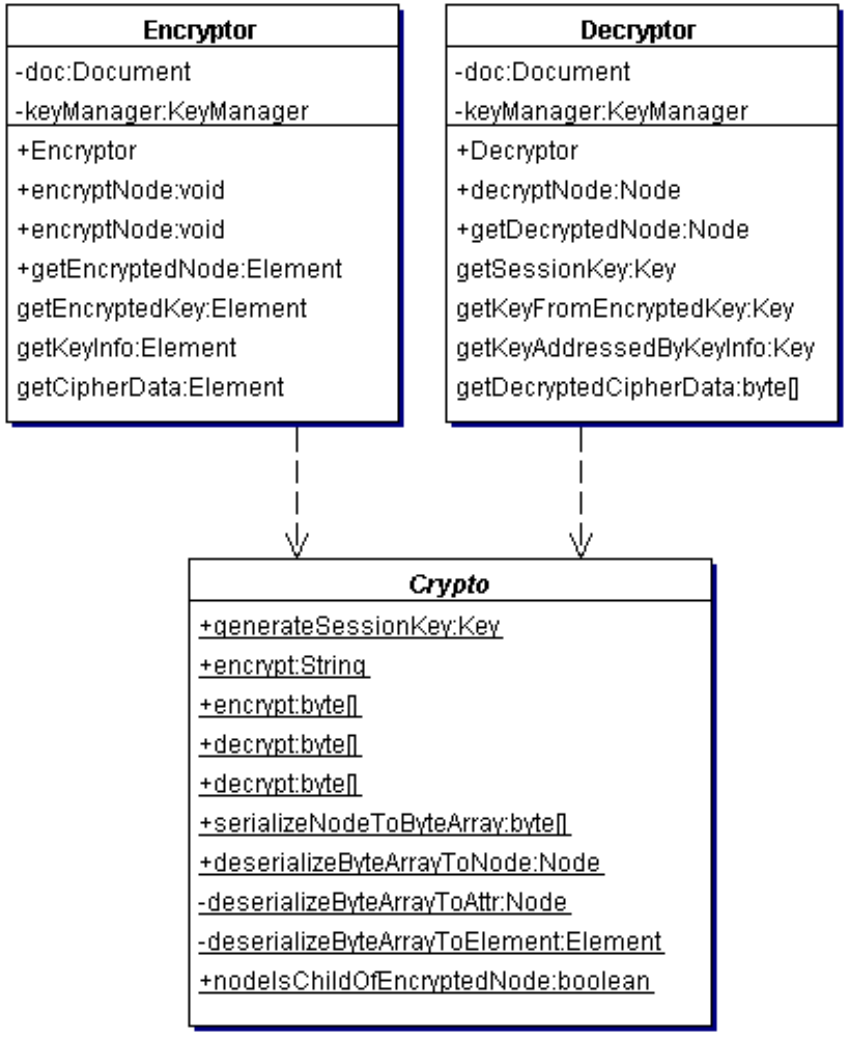


Paket-Übersicht



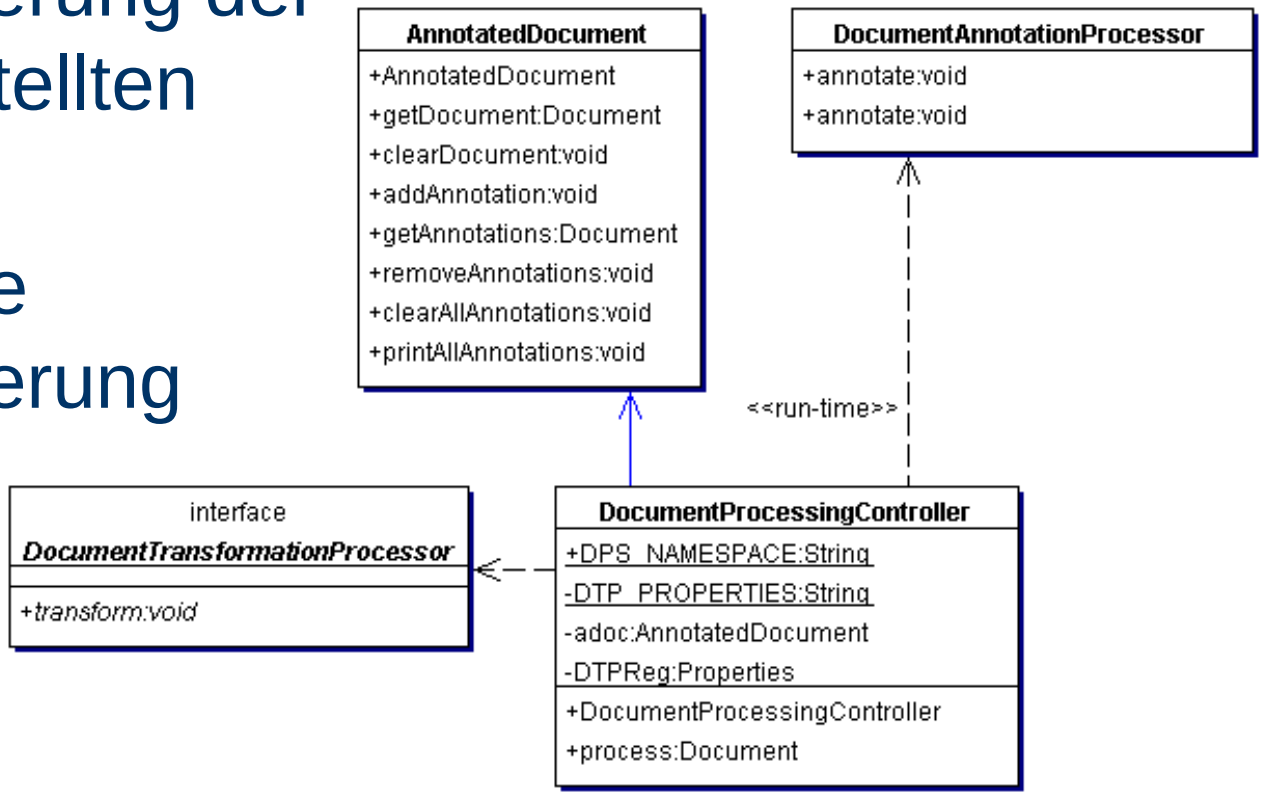
Verschlüsselung von XML-Daten

- Prototypische Implementierung der hier vorgestellten Spezifikation
- Die Möglichkeiten des KeyInfo-Elements werden nicht voll unterstützt
- Unabhängig vom DPF



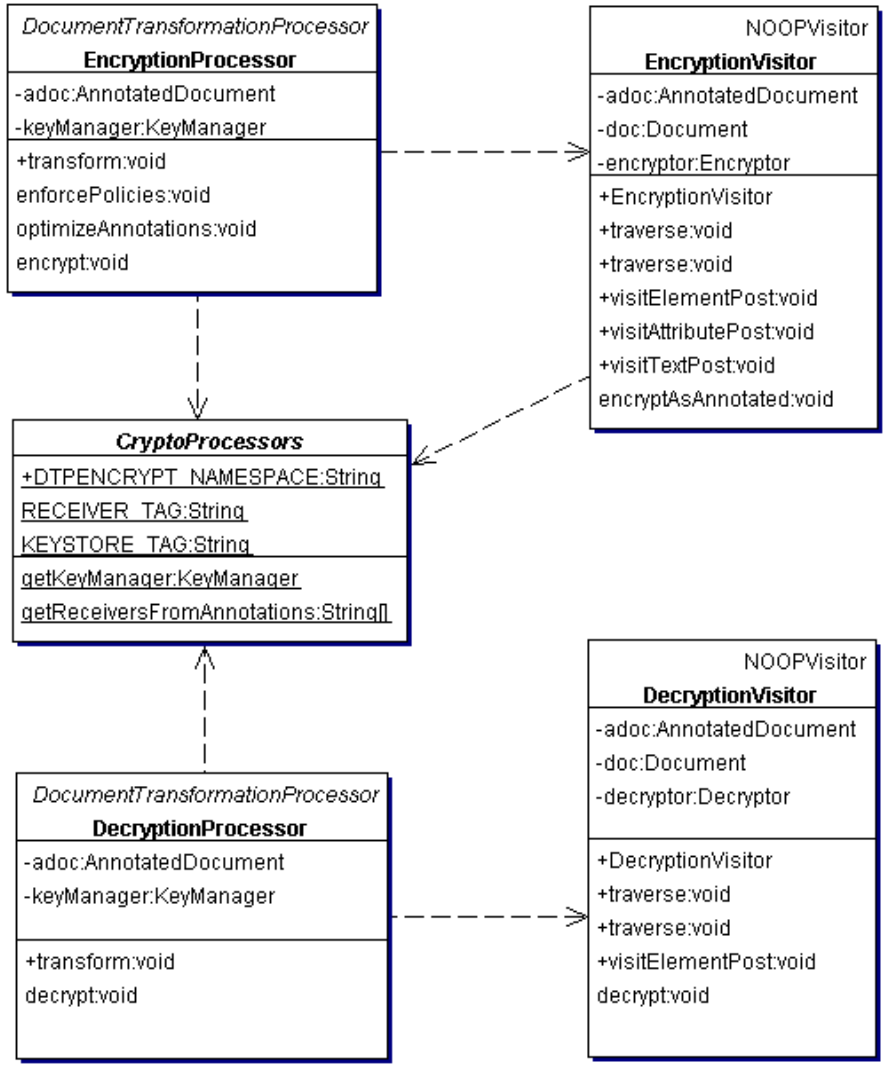
Document Processing Framework

- Prototypische Implementierung der hier vorgestellten Architektur
- Vollständige Implementierung



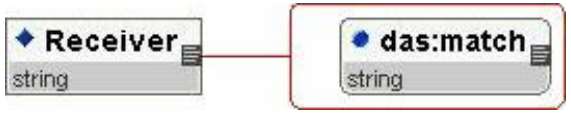
DTPs zur Ver- und Entschlüsselung

- Prototypische Implementierung der hier vorgestellten Architektur
- Die Unter-Komponenten zur Ver- und Ent-schlüsselung sind implementiert



DTP-Encrypt

- URN des Prozessors
  `urn:ipsi-gmd-de:DTP-Encrypt`
- Argument im DPS:
  Lokation des Schlüsselspeichers als URL
  `<keystore>./keystores/certificates</keystore>`
- Annotationen
  - XML-Namensraum
    `http://www.darmstadt.gmd.de/2001/04/DTP-Encrypt`
  - XML-Schema

    `das:match` ist aus dem XML-Schema für DAS importiert

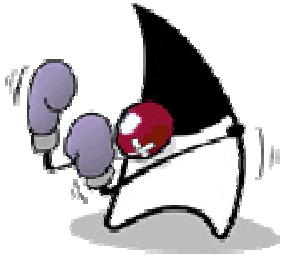
DTP-Decrypt

- URN des Prozessors
  `urn:ipsi-gmd-de:DTP-Decrypt`
- Argument im DPS:
  Lokation des Schlüsselspeichers als URL
  `<keystore>../keystores/MyPrivateKeys</keystore>`
- Annotationen
  werden nicht benötigt
- Entschlüsselt immer rekursiv und maximal

Probleme mit JCA/JCE (1)

- JCA/JCE unterspezifiziert
- Vergleich der Parameter für DES

Security Provider	Methode: javax.crypto. KeyGenerator. getInstance (algo)		javax.crypto. KeyGenerator. init(keysize)		java.security. Key. getAlgorithm()	
Eingabe/Ausgabe	algo="DESede"	algo="DESEDE"	keysize=168	keysize=192	"TripleDES"	"DESede" "DESEDE"
Baltimore KeyTools Lite v5 for Java	✓	✗	✓	✓		✓
Bouncycastle Crypto Package 1.0.5	✓	✓	✗	✓		✓
Cryptix JCE preliminary version 20000905	✓	✗	✓	✗	✓	
OpenJCE 1.1	✓	✗	✓	✓		✓
Sun JCE 1.2.1	✓	✓	✓	✗		✓



Probleme mit JCA/JCE (2)

- Vergleich der Parameter für RSA

Methode / Security Provider \ Eingabe/Ausgabe	javax.crypto. KeyGenerator. getInstance(algo) algo="RSA//PKCS1"	algo="RSA//PKCS#1"	algo="RSA//PKCS1Padding"	algo="RSA/ECB/PKCS1"	algo="RSA/ECB/PKCS#1"	algo="RSA/ECB/PKCS1Padding"	java.security. Key. getAlgorithm() "RSA"
Baltimore KeyTools Lite v5 for Java	✓	✓	✓ *1	✗	✗	✗	✓
Bouncycastle Crypto Package 1.0.5	✗	✗	✓	✗	✗	✓	✓
Cryptix JCE preliminary version 20000905	✗	✗	✗	✗	✓	✗	✓
OpenJCE 1.1	✗	✗	✗	✗	✗	✓	✓
Sun JCE 1.2.1	na	na	na	na	na	na	na

*1) javax.crypto.IllegalBlockSizeException: PKCS5::pad(byte[], int)
- this method is not implemented. Use PKCS_5.pad(...) instead.

na: Nicht verfügbar

Probleme mit XML-Technologien (1)

- Xerces und XML-Schema
  - Werden URIs als Datentyp spezifiziert, werden nur absolute URLs bei der Validierung akzeptiert
  - Relative URLs in den XML-Schemas werden in bestimmten Fällen nicht korrekt aufgelöst (Import von XML-Schemas innerhalb eines XML-Schemas)
- Serialisierung von DOM-(Teil-)Bäumen: Überall Fehler bei XML-Namensräumen
- Absoluten URL eines XML-Dokuments berechnen ist aufwändig

Probleme mit XML-Technologien (2)

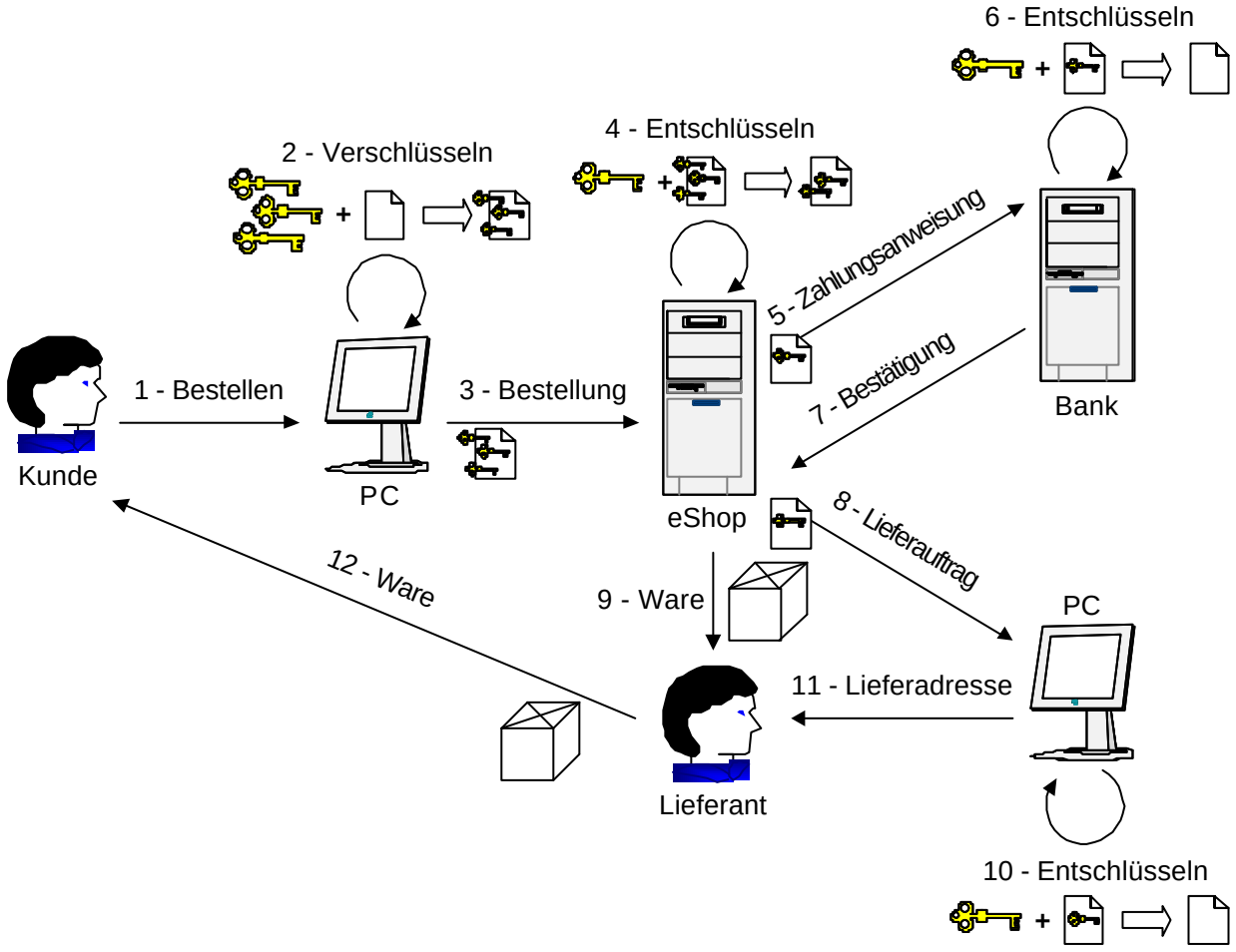
- XPath-Ausdrücke und XML-Namensräume
  - Die Bindung der Namensräume erfolgt nicht über deren URIs sondern über Präfixe – diese sind aber variabel
  - Joker sind für Präfixe nicht erlaubt
  - Relativ komplexe Ausdrücke werden notwendig

```
//*[local-name()='inherit' and
    namespace-uri()='http://examples.org/anotherns']
```

  - Zusätzlich unerwartete Ergebnisse bei der Selektion von Elementen mit vererbtem Namensraum (Liegt wahrscheinlich am verwendeten Xalan)

Demonstration

Szenario „Anonymisierte Bestellung“



Ausblick

Ausblick (1)

- Vervollständigung der Implementierungen
- DTP-Decrypt um eine Unter-Komponente zur Überprüfung von Sicherheitsrichtlinien erweitern
- Gewährleistung der Vertrauenswürdigkeit durch Signaturen (XML Signature) mittels eines eigenständigen DTPs und/oder einem integrierten DTP
- Spezifikation/Implementierung weiterer DTPs
  z.B. XML Encryption (W3C), Policy-Checker, ACLs, XSLT, Messaging, ...

Ausblick (2)

- Übergabe von Parametern an DPS- und/oder DAS-Dokumente über den DPC
- Prozessoren um Rückgabewerte erweitern und eine Sprache zur Steuerung des Kontrollflusses in DPS-Dokumenten einführen
- Grafische Werkzeuge zur Erstellung / Pflege von DPS- und DAS-Dokumenten
- Komponente zur Assoziation von XML-Dokumenten mit den dazu relevanten DPS-Dokumenten – kann dann einen vollständigen Dienst zur Transformation von Dokumenten bieten

Ausblick (3)

- Unterstützung grosser dynamischer Empfängergruppen und sicherer Workflow-Systeme durch extern referenzierte Sitzungsschlüssel

Diskussion

Diskussion

Quellen

- Diplomarbeit (PDF)
  http://www.informatik.uni-frankfurt.de/~priewe/
- Arne Priewe
  eMail: priewe@informatik.uni-frankfurt.de
- Gerald Huck
  eMail: huck@darmstadt.gmd.de