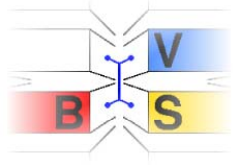


Datenschutz für medizinische Patientendaten

Thomas Scheffler
scheffler@cs.uni-potsdam.de

Universität Potsdam
Lehrstuhl Betriebssysteme und Verteilte Systeme

18. Februar 2009



Inhalt



Grundlagen

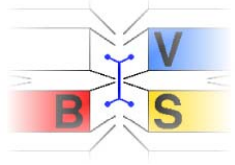
- Motivation und Ziel der Arbeiten

Technische Umsetzung

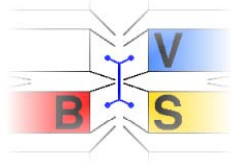
- Architektur der Lösung
- Policy Enforcement Point auf Basis des JavaSecurity Frameworks
- Realisierbare Zugriffsbeschränkungen durch Java Permissions

Usability Aspekte

- Anforderungen an die Erstellung von Datenschutzrichtlinien
- Realisierung priorisierter Teil-Richtlinien



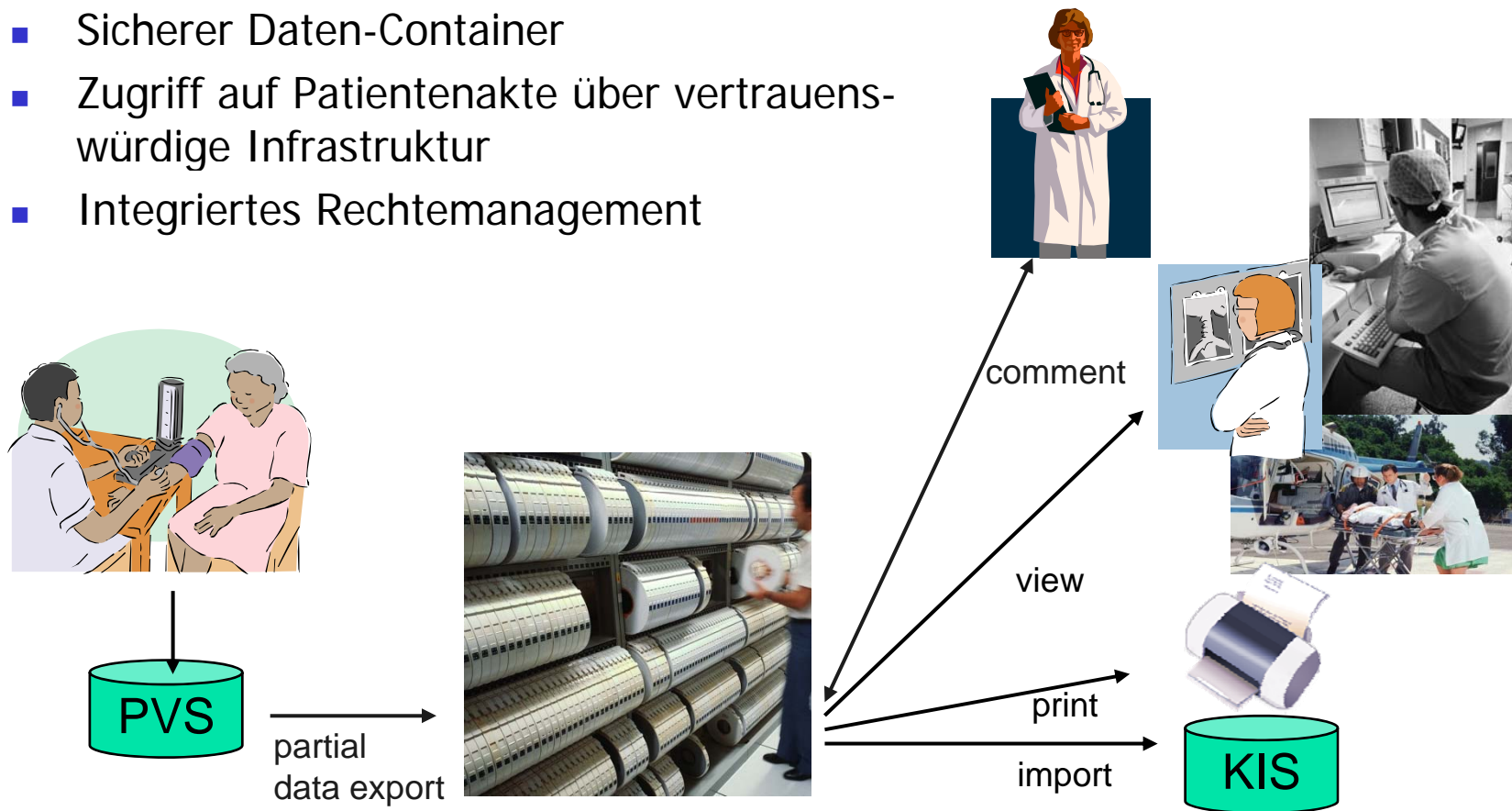
Motivation

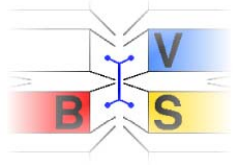


Elektronische Patientenakten



- Speicherung von Teilen der realen Krankengeschichte
- Sicherer Daten-Container
- Zugriff auf Patientenakte über vertrauenswürdige Infrastruktur
- Integriertes Rechtemanagement

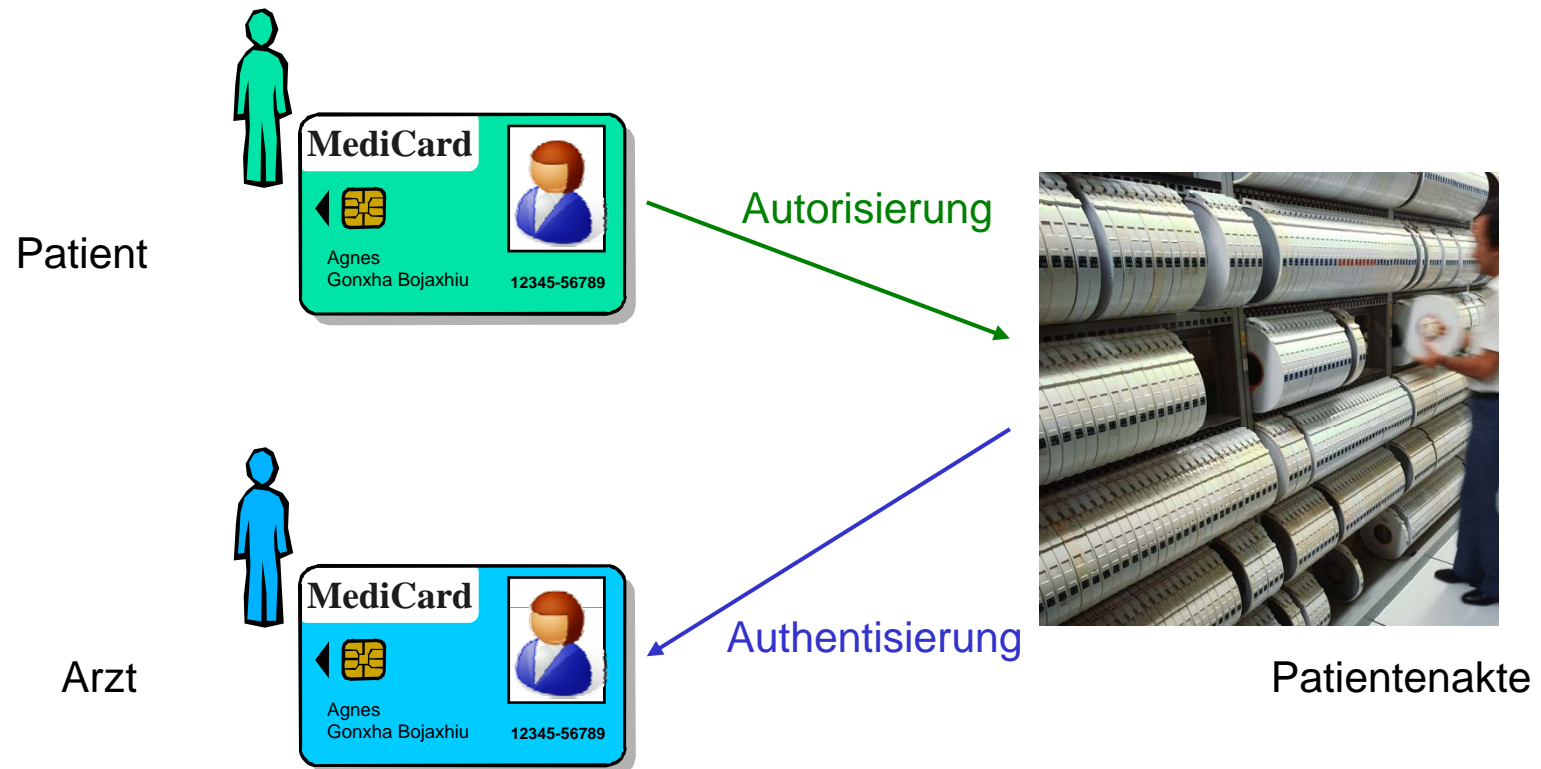


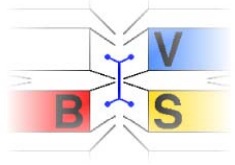


Elektronische Patientenakten



- Patienten autorisieren Zugriff auf Daten der Patientenakte
- Ärzte müssen sich für den Zugriff auf Patientendaten authentisieren





Elektronische Patientenakten

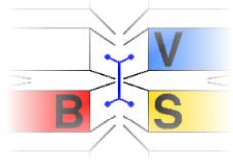


Offene Fragen zu Zugriff und Nutzung:

- Auf welche Daten kann der Arzt zugreifen?
 - Kann er alle Daten in der Patientenakte sehen?
- Wie lange ist der Zugriff möglich?
- Können Daten aus der Patientenakte in PVS und KIS importiert werden?
- Wie kann der Zugriff effektiv widerrufen werden?
Was passiert mit den Daten, auf die Zugriff Bestand?
- Wie kann zweckfremde Nutzung verhindert werden?
- Welche Rechte haben weitere Akteure (Pfleger, Kassen, ...)

Wie läßt sich eine einfache und wirkungsvolle Rechteverwaltung realisieren?





Besitzerkontrollierter Zugriff auf Daten

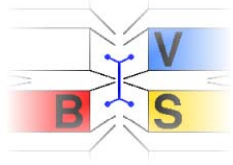


- Der **Datenbesitzer** legt die Zugriffspolicy für die Daten fest



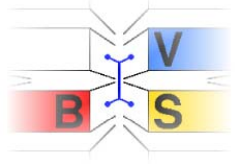
- Verteilter **Reference Monitor** überwacht die erlaubte Verwendung der Daten

Datenschutz = Zugriffskontrolle + Nutzungskontrolle

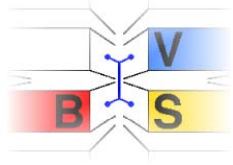


Frage: Wie können sensitive Daten in mobilen, elektronischen Patientenakten geschützt werden - angesichts des gewollten Zugriffs durch verschiedene Akteure?

- Patientendaten werden als semi-strukturierte XML-Dokumente gespeichert
 - Daten werden zusammen mit ihren Zugriffsrechten vorgehalten
- Distributed Access-Control Framework
 - Jegliche Zugriffe auf Daten werden gegenüber der Policy in Echtzeit evaluiert
- Automatisierte Durchsetzung von Zugriffsrechten für beliebige Anwendungen



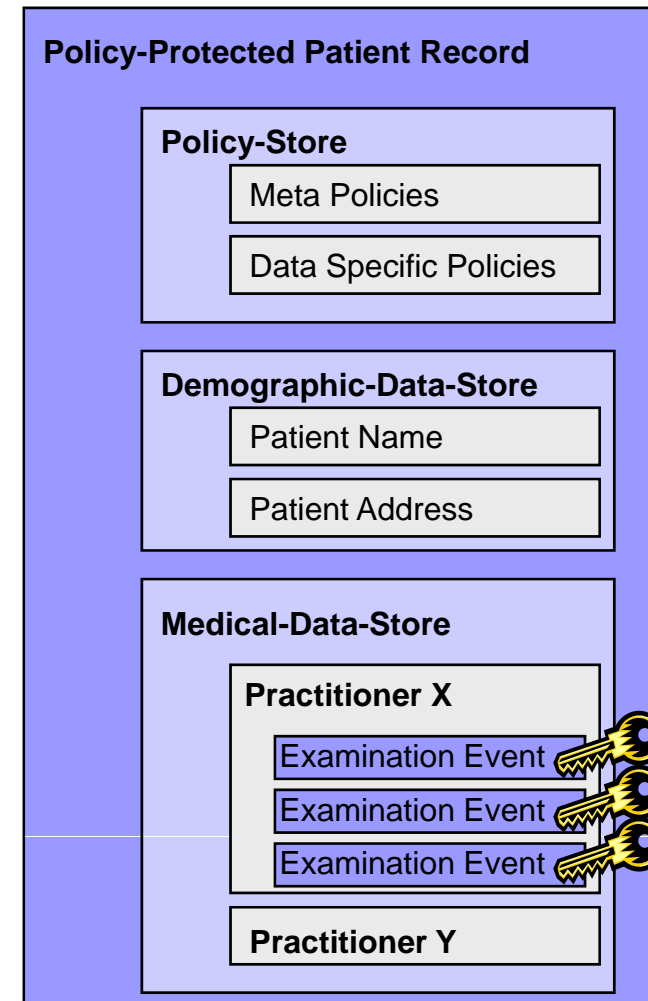
Technische Umsetzung

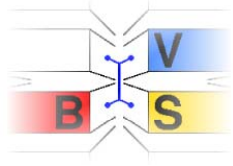


Datenmodell Patientenakte



- Der **Policy-Store** ist Speicherort für
 - Generische Policies
 - Nutzergenerierte Policies
- Im **Demographic-Data-Store** sind allgemeine Daten zum Patienten abgelegt
- Im **Medical-Data-Store** sind die medizinischen Daten zu Untersuchungen und Behandlungen abgelegt

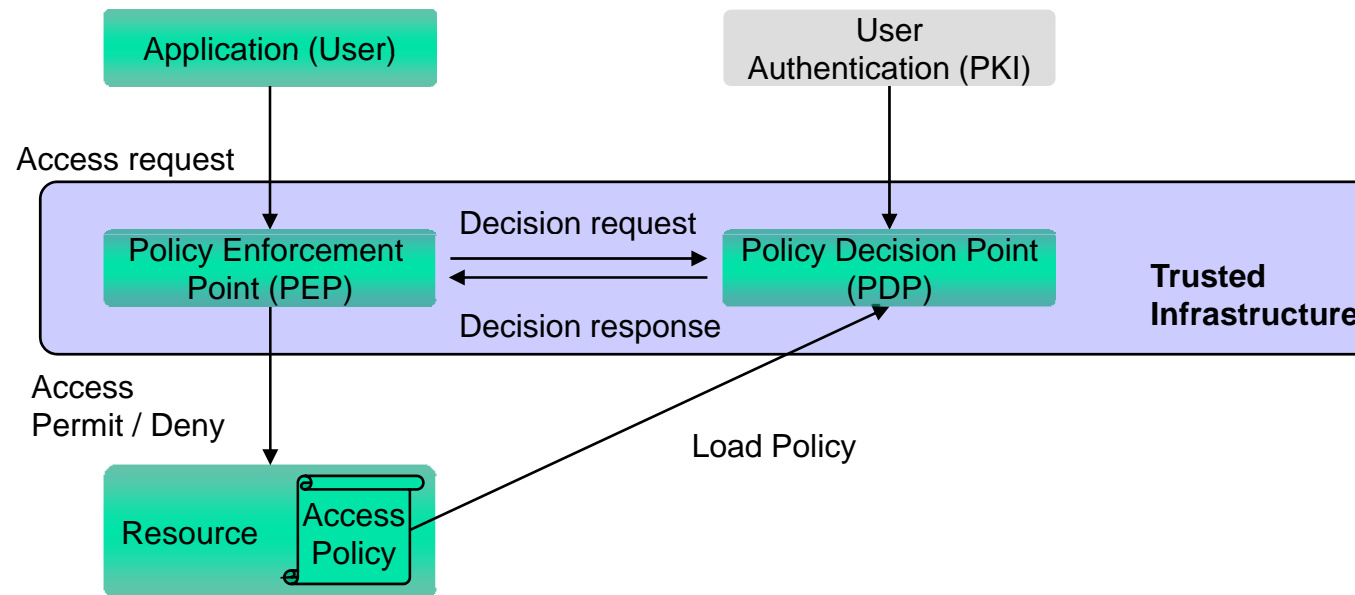


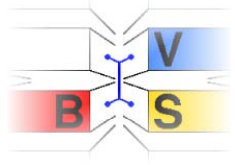


Prototypische Realisierung

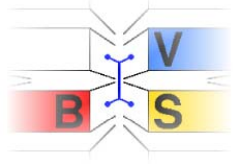


- **Grundidee:** Übersetzung von XACML Policies in Java Permissions, welche vom Java SecurityManager zur Laufzeit überwacht und durchgesetzt werden.
 - Ermöglicht die Trennung in vertrauenswürdige Infrastruktur-Komponenten und nicht-vertrauenswürdigen Anwendungen





Usability Aspekte



Erstellung von Zugriffsrichtlinien

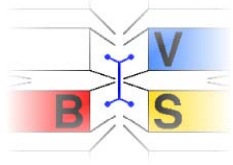


Einige Herausforderungen für die Policy-Erstellung:

- Alle Untersuchungsdaten müssen durch Richtlinien geschützt werden
 - Bereitstellung einer Default-Policy, die standardmäßig ein bestimmtes Sicherheitsmaß garantiert, ohne dass Anpassungen erfolgen müssen

- Benutzer sollen absolute Hoheit über die Policy-Erstellung haben
 - Benutzer können die Default-Policy entsprechend ihrer Bedürfnisse jederzeit anpassen
 - Bestimmte Nutzerrechte für die Daten-Autoren (Ärzte,...) müssen gewahrt bleiben

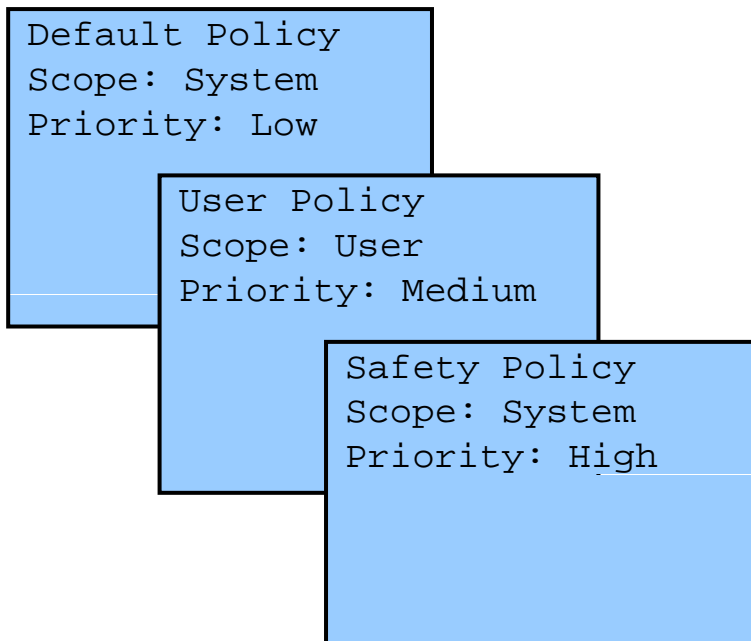
- Benutzer sind keine XACML und Sicherheitsexperten
 - Fehleingaben können vorkommen, der Nutzer muss ggf. vor sich selbst geschützt werden



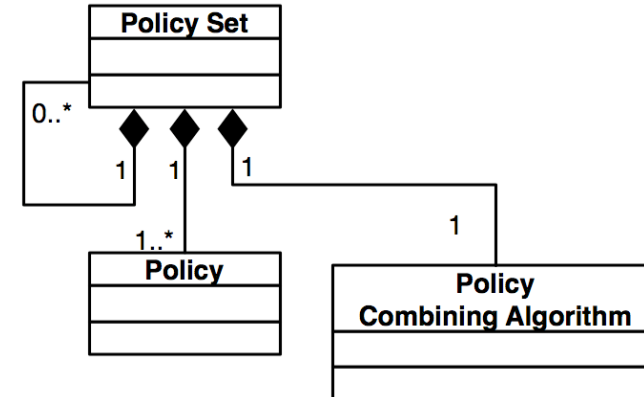
Priorisierung der Policies

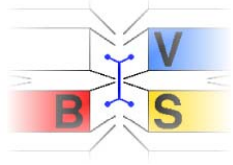


- Die geltende Datenschutzrichtlinie setzt sich aus der Vereinigung von 3 Teil-Richtlinien zusammen, wobei nur die User-Policy editierbar ist
- Nutzung eines **XACML-PolicySet** mit angepasstem **Combining Algorithmus**



XACML: Policy Combining Algorithmus



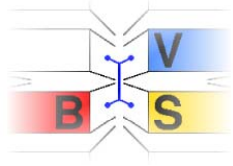


Zusammenfassung



- Implementierung von nutzerkontrollierten Datenschutzmechanismen ist machbar
 - Nutzung von XACML Policies für die Beschreibung
- Durchsetzung der Policies erfordert vertrauenswürdige Infrastruktur
 - Java Security Framework kann angepasst werden, bestimmte Nutzungsszenarien automatisch durchzusetzen
- Funktionsfähiger Prototyp kann im Foyer besichtigt werden!

Fragen?



Thomas Scheffler
Betriebssysteme und Verteilte Systeme
Universität Potsdam

scheffler@cs.uni-potsdam.de
www.cs.uni-potsdam.de