TeleTrusT – Bundesverband IT-Sicherheit e.V.

Der IT-Sicherheitsverband e.V.

TeleTrusT-interner Workshop

Nürnberg, 21./22.06.2012

Peter Trommler
Georg-Simon-Ohm-Hochschule Nürnberg

Defence in Depth: Fein-granulare Zugriffskontrolle in der Datenbank anhand eines Projekts aus der Pflegeunterstützung

Übersicht

ISSECO®
SECURE SOFTWARE ENGINEERING

- Internetbasierte Pflegeunterstützung: VAMOS
- Herausforderung Zugriffskontrolle
- Modellierung
- Implementierung, technisch
- Umsetzung, geschäftlich

VAMOS: Internetbasierte Pflegeunterstützung

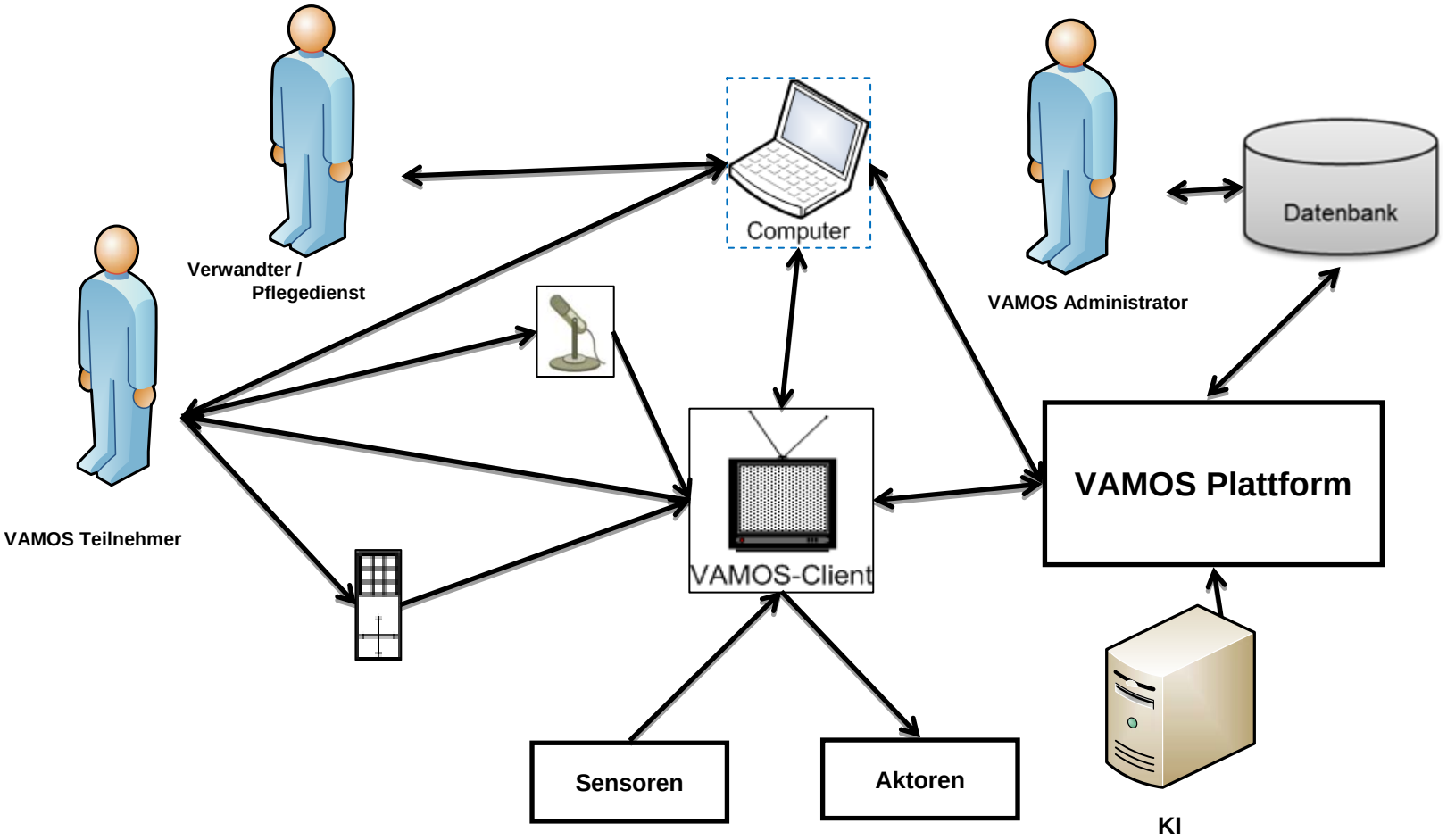
ISSECO®
SECURE SOFTWARE ENGINEERING

- **VAMOS** (Versorgungseffizienz durch assistive, modulare Technologien in bedarfsorientierten Szenarien)
- **Ziel des Projekts**
  - Unterstützung pflegebedürftiger Menschen
  - Senioren dauerhaften Aufenthalt im eigenen Heim ermöglichen
- Entwicklung einer Plattform
  - Anwendungsszenarien (Sensoren, Aktoren)
  - Hilfsfunktionen (Kalender, Video-Telefonie, …)
- Gefördert vom Bundesministerium für Bildung und Forschung (16KT0943)

Bundesministerium
für Bildung
und Forschung

VAMOS: Übersicht

ISSECO®
SECURE SOFTWARE ENGINEERING



VAMOS: Zugriffskontrolle

ISSECO®
SECURE SOFTWARE ENGINEERING

- VAMOS Teilnehmer
  - Zugriff auf persönliche Daten
- VAMOS Fernseher
  - Daten der angeschlossenen Sensoren schreiben
- Pflegedienst, Arzt
  - Daten eigener Patienten zugreifen
- Künstliche Intelligenz
  - Zugriff auf anonyme Daten
- Verwandter
  - Vom VAMOS Teilnehmer freigegebene Daten einsehen

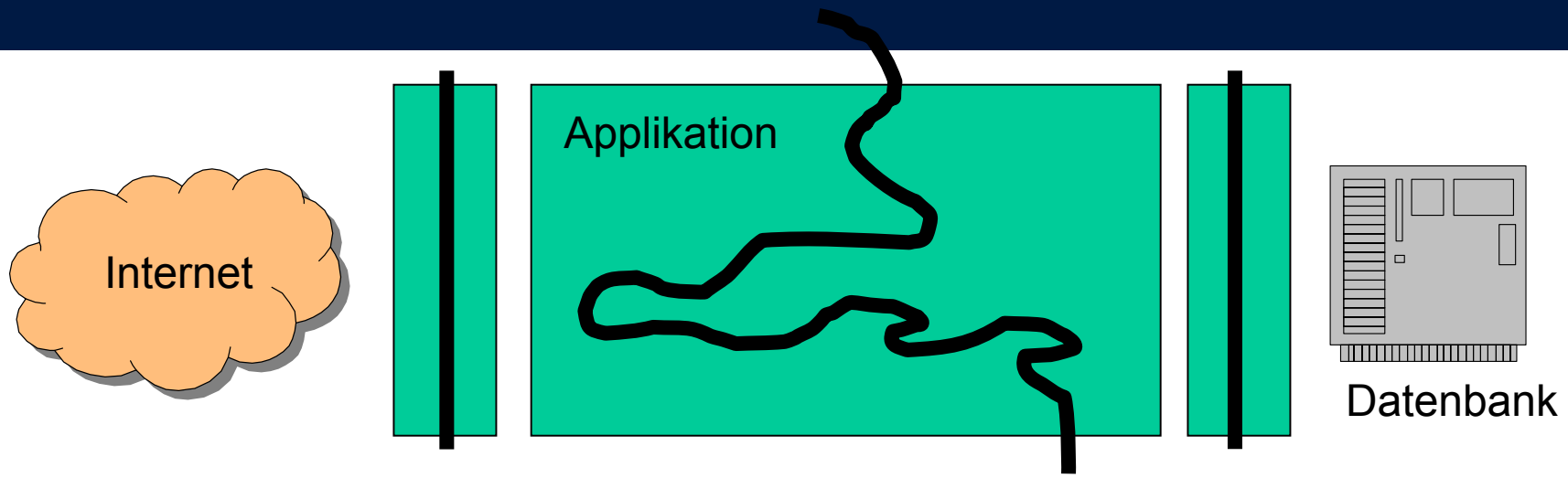
Herausforderung

ISSECO®
SECURE SOFTWARE ENGINEERING

- Zugriffskontrolle über die gesamte Anwendung verteilt
- Herausforderungen
  - Vollständigkeit
  - Korrektheit
  - Verständnis
- Abhängigkeit von anderem Code
- Lösung: Trenne Zugriffskontrolle vom Anwendungscode

Motivation

ISSECO®
SECURE SOFTWARE ENGINEERING



- **Wo Zugriffskontrolle implementieren?**
- Kriterien
  - Datenbank zentrale Komponente
  - keine Zusatzprogramme (WAF, ...)
  - geringe Änderungen im Webcode

Implementierung

ISSECO®
SECURE SOFTWARE ENGINEERING

- Parameterized Views (Roichman, Gudes; 2007)
  - View hängt von Parametern ab
  - Session-Identifier als Parameter
- Implementierung Parameterized Views
  - User Defined Functions
  - Query Rewriting
  - Oracle's Virtual Private Databases
- Code Generator
- Erweiterung der Query um Session-Parameter
  - muss durch Applikation durchgezogen werden
  - wird in Abfrage eingefügt

Modellierung der Zugriffskontrolle

ISSECO®
SECURE SOFTWARE ENGINEERING

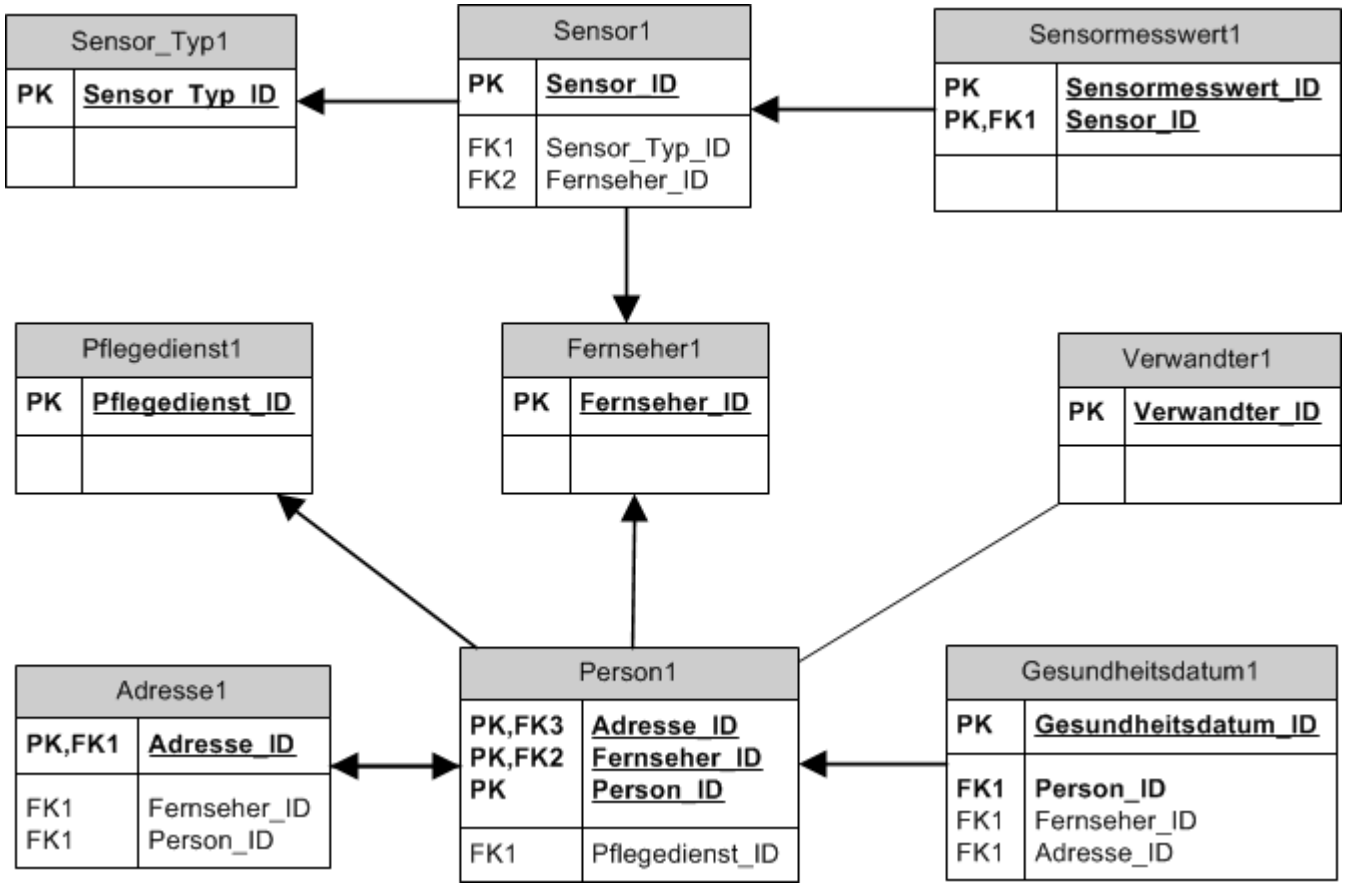
- Benutzer fallen in Kategorien
- Modell für Zugriffsregeln
  - eigene Daten
  - Aktionen in der Vergangenheit
  - allgemeine Regeln

➔ Zugriffskontrolle muss fein-granular arbeiten

- Beschränkung auf Spalten **und** Zeilen
- Abhängigkeit vom Inhalt der Datenbank

Modellierung „Eigene Daten“

ISSECO®
SECURE SOFTWARE ENGINEERING



Beispiel: VAMOS Teilnehmer

ISSECO®
SECURE SOFTWARE ENGINEERING

Befehlsaufbau

```
Anchor: Tabelle_A.Tabelle_A_ID =
  EXT_ID

Tabelle_A -> Tabelle_B
  VIA Tabelle_B.Tabelle_A_ID

Tabelle_B: Keyword;
```

Abfrage Gesundheitszustand

```
Anchor: Person.Person_ID =
  userid;

Person -> Gesundheitsdatum
  VIA
  Gesundheitsdatum.Person_ID;

Gesundheitsdatum: read only;
```

Beispiele: Fernseher und Verwandter

ISSECO®
SECURE SOFTWARE ENGINEERING

Fernseher empfängt Messwerte

```
Anchor: Fernseher.Fernseher_ID =
  ID(CERTIFICATE);

Fernseher -> Sensor
  VIA Sensor.Fernseher_ID;
Sensor -> Sensormesswerte
  VIA Sensormesswerte.Sensor_ID;

Sensormesswerte: create;
```

Verwandter greift auf Gesundheitszustand zu

```
Anchor: Verwandter.Verwandter_ID
  = userid;

Verwandter -> Person
  VIA Verwandter.teiln <->
      Person.view;
Person -> Gesundheitsdatum
  VIA Gesundheitsdatum.Person_ID
  AND Clearance(Person_ID,
        Verwandter_ID, level);

Gesundheitsdatum: read only;
```

Geschäft oder Sicherheit?

ISSECO®
SECURE SOFTWARE ENGINEERING

- Liste der Konten eines Kunden anzeigen
  - Es fehlt ein Konto: Geschäft
  - Ein Konto zu viel: Sicherheit
- Geschäftsregeln haben Sicherheitsanteil
- Herausforderung: Herausarbeiten der Sicherheitsanforderungen

Business Rules Management System

ISSECO®
SECURE SOFTWARE ENGINEERING

- Wer definiert die Sicherheitsregeln?
  - Fachabteilung
  - IT-Abteilung
  - Gemeinsam in Workshops
- Neue Rolle für die Fachabteilung
  - Neue Verantwortung
  - Enge (und gute) Zusammenarbeit mit IT-Abteilung

Access Control Engineering

ISSECO®
SECURE SOFTWARE ENGINEERING

- Identifiziere Zugriffskontrollanforderungen früh im Projekt
- Verfeinere mit der Verfeinerung der funktionalen Anforderungen
  - (Teil-)Automatisierung der Verfeinerungsschritte
  - Verfeinerung auf Korrektheit überprüfen
  - Audit der Regeln (manuell)

Erfahrungen aus der “Praxis”

ISSECO®
SECURE SOFTWARE ENGINEERING

- “Saša, Autorisierung machst Du, dann mach’ ich da also nichts mehr.”
- DACH Security 2011: “Dafür bekommen Sie von mir kein Geld!“
- Spezifikation Schutzbedarf der Daten
  - “Das ist mir zu technisch, da klink’ ich mich aus.”
  - Mit/ohne SSL, mit/ohne Verschlüsselung, …

Ausblick

ISSECO®
SECURE SOFTWARE ENGINEERING

- Security Requirements Engineering
- Integration in den Softwareentwicklungsprozess
- Effizienz der Implementierung
  - Caching der parameterized Views
- Graphische Modellierung
  - BPMN und UML

Literatur

ISSECO®
SECURE SOFTWARE ENGINEERING

- Rizvi et. al.: Extending query rewriting techniques for fine-grained access control. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, S. 551-562.
- Roichman und Gudes: Fine-grained Access Control to Web Databases. In Proceedings of SACMAT ’07, 2007, S. 31-40.
- Prijovic und Trommler: Zugriffskontrolle in der Datenbank: Vamos eine Fallstudie. In Schartner, Taeger (Hrsg.) DACH Security 2011, pp. 147-156, syssec Verlag.
- Trommler: A Model For Fine-grained Access Control to Web Databases: VAMOS – A Case Study. Angenommen für TPMC 2012, Lissabon, Portugal.
- Rossel, Grosse, Prijovic und Trommler: Zugriffskontrolle in Webdatenbanken mit Query Rewriting. Angenommen für DACH Security 2012, Konstanz.