

# Verification of Access Control in Softwaresystems

## Bachelor Thesis

Julian Hinrichs | Advisor: M.Sc. Stephan Seifermann

Reviewer: Prof. Dr. Ralf H. Reussner | ~~Prof.~~ Jun.-Prof. Dr.-Ing. Anne Koziolk

SOFTWARE-ENTWURF UND -QUALITÄT  
INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION, FAKULTÄT FÜR INFORMATIK

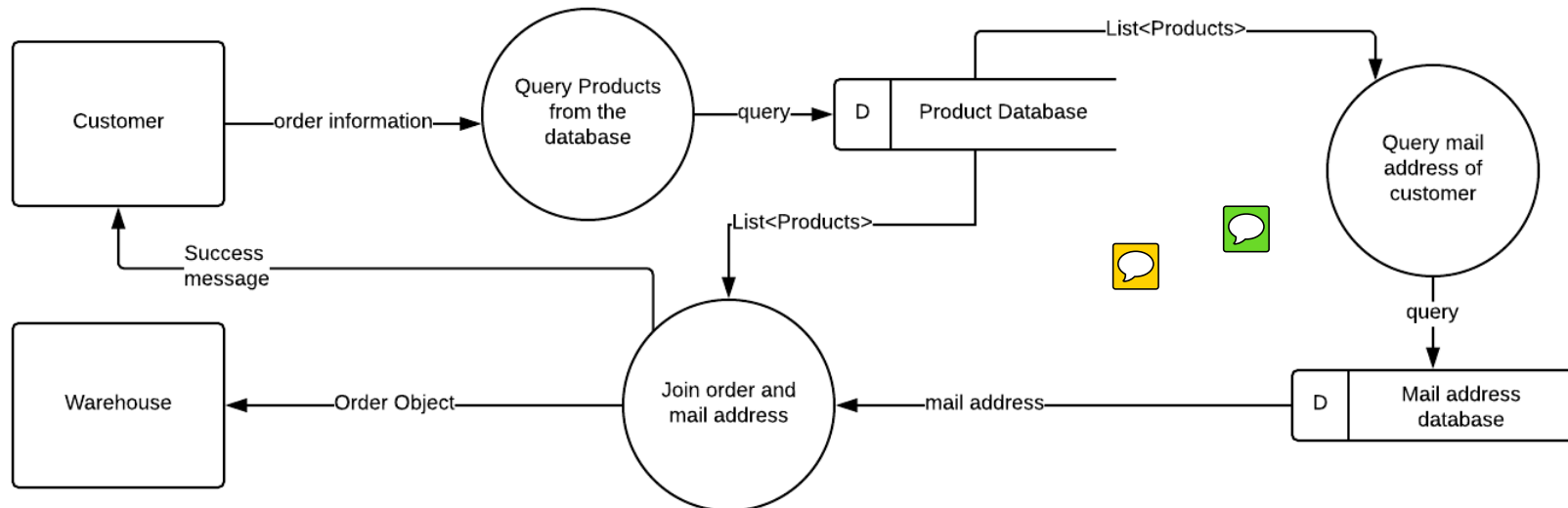


# Motivation

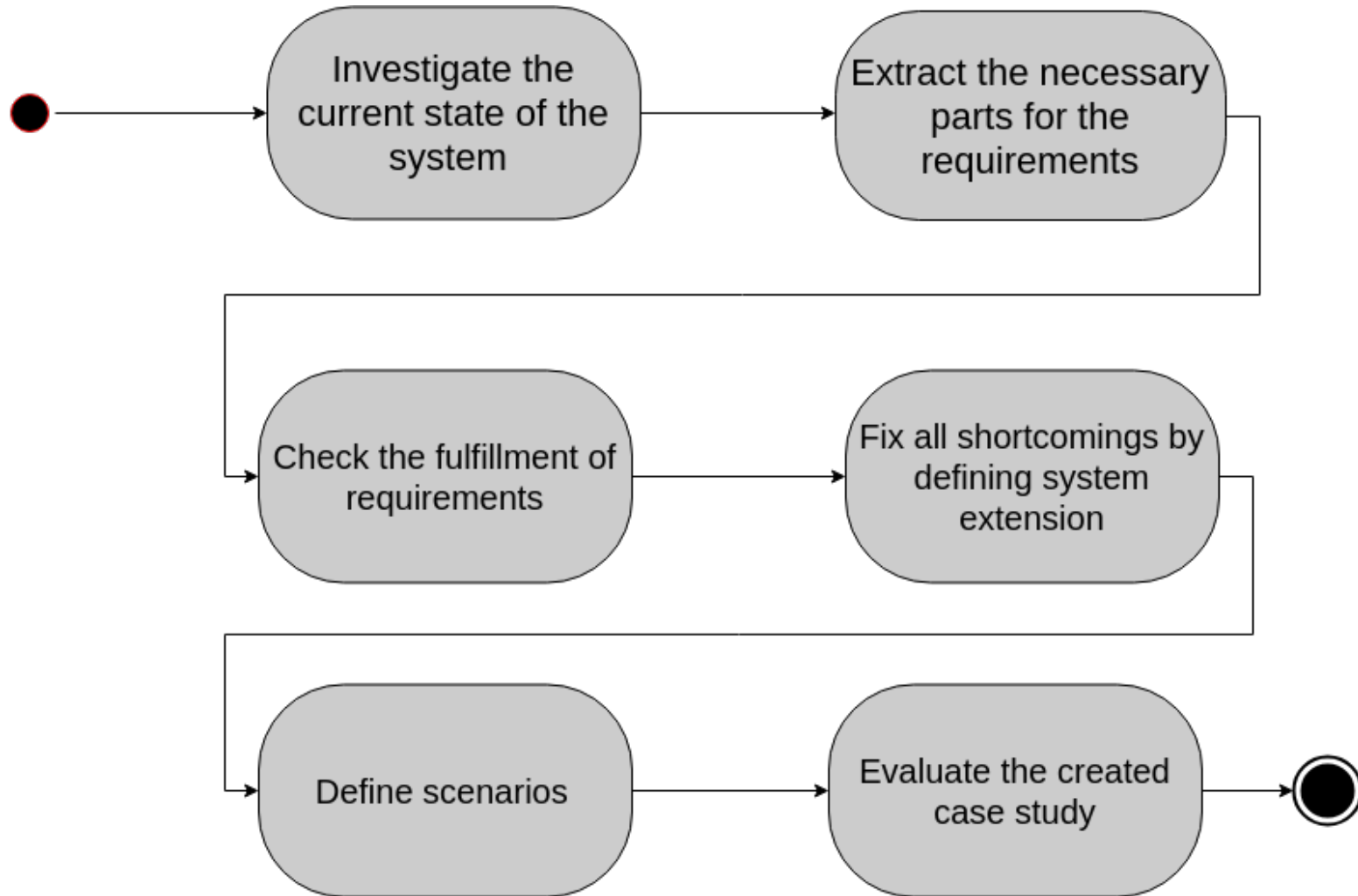
- Security and privacy becomes more important, ~~especially privacy.~~
- Ensure compliance of privacy on an architectural level.
- Different approaches
  - Data-based anylsis → Databased privacy analysis
  - Attacker model → UMLSec
  - Modeling of a safe state → SecureUML
- Databased approaches usually using case studies for evaluation
- Presentation of a procedure to create viable case studies and evaluating the quality.

# Foundations

- Security relevant data
  - Data that is worth protecting from threats from the outside
- Data flows
  - Describe the movement of data and the changes to data in a system



# Procedure overview



# Procedure: requirements for a case study

R1	Component based system
R2	Definition of use case
R3	Security relevant data
R4	Definition of user roles
R5	Definition of access rights
R6	Definition of the type of data processing





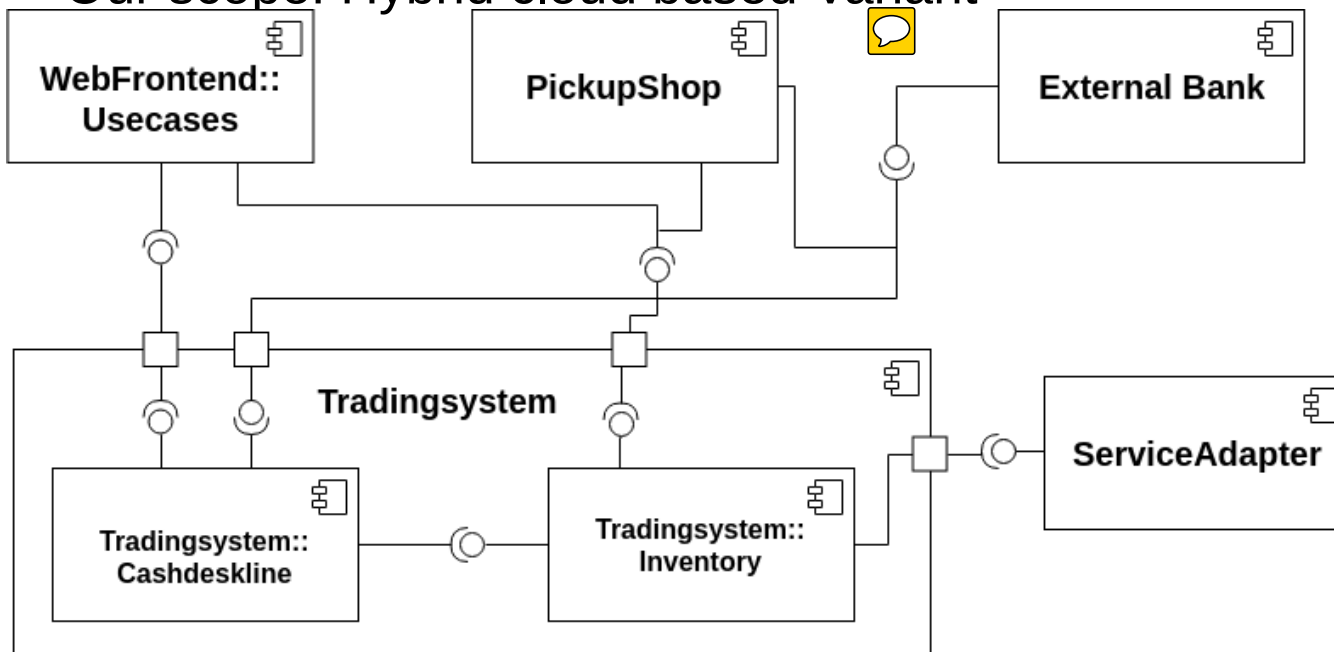
# Procedure: shortcomings and milestone

- Check the fulfillment of the requirements for a case study and create a list of shortcomings
- Fix all the shortcomings, for example, adding extensions to the system
  - the milestone is reached, the system is in a state so it is possible to create a case study
- Define different scenarios:
  - Scenarios are used to model typical interactions with the given system
  - From the scenarios the concrete data flows for the system are derived
- Evaluation
  - Quality of access rights
  - Covered information flow classes



# CoCoME overview

- CoCoME abstracts ~~the~~ a supermarket system
  - Managing the stock
  - Selling of products
- Our scope: Hybrid cloud based Variant




# Fulfillment of requirements R1-R4


- R1: component based system → check
- R2: definition of use cases → check, 13 use case are defined in various publications
- R3: Security relevant data
  - Account data: security relevant
  - Customer data: security relevant
  - System data: security relevant
  - Product & sales data: security relevant if informations for security relevant classes may be derived.
- R4: user roles
  - 5 different roles defined in the documentation
  - Customer, cashier, StockManager, StoreManager, EnterpriseManager





- R5: access rights

- For each user and each component the access rights are defined for each data type
- Four levels of access rights:
  - Level 1: Full Access
  - Level 2: Access to used data 
  - Level 3: Access to owned data
  - Level 4: default/no access

	Webfront-end	TS:inventory:app-lication	TS:inventory:data
Stock Manager	Customer data : 4 Account data : 3 P & s data : 2 System data : 4	Customer data : 4 Account data : 3 P & s data : 2 System data : 4	Customer data : 4 Account data : 3 P & s data : 2 System data : 4 

# R6: types of data processing



- Define for each component and each datatype the possible operations
- Three types of data processing:
  - Transmission of data
  - Perform relational algebra operations
  - I/O operations



	Customer data	Account data	Product & sales data	System data
Webfrontend	transmit	transmit	I/O operations transmit	Non-existent



# Definition of scenarios

- Mostly **oriented on** the use cases or access rights
- **The goal is to model interaction with the system**
- Scenarios allow to get an detailed insight in the system under **investigation**
- A **desirable** goal is to cover all information flows with scenarios to be able to check Non-influence (Non-interference + non-leakage). 
- Four information flow classes
  - Illegal information flow
  - Observable information flow
  - Flow from high to low 
  - Direct information flow

# Scenario: StockManager requests the report for a customer



# Evaluation: Case Study



Access Rights	Fulfilled ?
Positive	check
Need-to-Know	check
Aspect-oriented	check

Information flow classes	Covered ?
Observable flow	X
Direct information flow	X
Flow from high to low	check
Illegal information flow	check



# Evaluation: Method and PCM extension

- Applicability of the Method

- Applicable for the examined excerpt of CoCoME



→ method applicable for the investigated and similar systems

- PCM



Relational algebra	check
I/O Operations	check
Transmission of data	check

Change of access rights	check
Alternation of data	check
Access control matrix in the model	X



- Only one type of system was investigated and from that system only an excerpt.
- We checked three out of seven available criteria due to:
  - Time constraints
  - System constraints
- Not all information flow classes are modeled
  - Two information flow classes are missing
- For each information flow class a violation and a non-violation should be modeled in the case study.



# Related Work



- Case studies are commonly used in other fields than computer science, like law, sociology, health care, etc.
- The methodology to create case studies in these fields is well documented and heavily investigated.
- In computer science, case studies are mainly used to investigate how different challenges were tackled.
- A **comparable** publication: Everend and Bögeholz
  - Evaluation criteria to measure good access rights
  - **A shorter version** for the creation of a case study, which we used **as comparison** for our process.



# Future Work

- Method:
  - Apply to other systems (travelsystem, etc)
- Case study
  - Using of the case study in a data based pivacy analysis.
  - Evaluate for the two additional criteria in the short term and all seven criteria in the long term
  - Additional scenarios for the examined excerpt to cover all information flow classes.
  - Define a violation and a non-violation for each information flow class in the scenarios
- PCM
  - Access Control matrix in the same model

# PIBA

- Problem

- Security, especially privacy becomes more important.



- Ensure **compliance** by evaluating privacy on an architectural level with a **data** based privacy analysis.

- **Data** based approaches uses case studies of a system for validation.

- Idea

- Introduction of a process to create usable case studies.

- Benefit



- Usable case studies for **data** based privacy analysis

- Approach



- Define requirements for a system so that it is possible to create a case study out of the system .



# Access control matrix

# Verification of Access Control in Softwaresystems

## Bachelor Thesis

**Julian Hinrichs | Advisor: M.Sc. Stephan Seifermann**

**Reviewer: Prof. Dr. Ralf H. Reussner | Prof. Jun.-Prof. Dr.-Ing. Anne Koziolk**

SOFTWARE-ENTWURF UND -QUALITÄT  
INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION, FAKULTÄT FÜR INFORMATIK



## Motivation

- Security and privacy becomes more important, especially privacy.
- Ensure compliance of privacy on an architectural level.
- Different approaches
  - Data-based analysis → Databased privacy analysis
  - Attacker model → UMLSec
  - Modeling of a safe state → SecureUML
- Databased approaches usually using case studies for evaluation
- Presentation of a procedure to create viable case studies and evaluating the quality.

Sicherheit, speziell Datenschutz immer wichtiger in den kommenden Softwaresystemen. Da es eine nicht funktionale Anforderung ist, kann man es schlecht überprüfen. Es gibt 3 Hauptideen

UMLsec → mit eine Angreifermodell schwächen in der Modellierung aufdecken

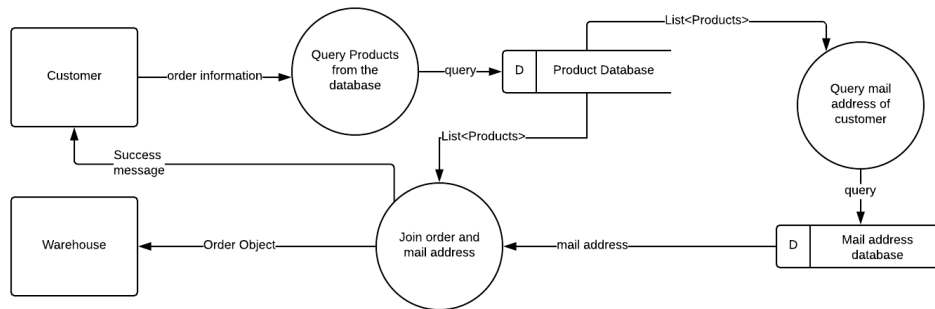
SecureUML → Systemmodell und Sicherheit in einem Modell speichern

Data based privacy analysis DBPA → mithilfe von Datenflüssen die Sicherheit auf Architektur Ebene überprüfen

Unser Beitrag setzt bei DBPA an, da diese Ansätze case studies von systemen verwenden. Es wurde ein Vorgehen entwickelt, um diese Case studies zu erstellen. Die erstellten case studies können noch zu anderen Zwecken verwendet werden, ist aber nicht im scope der thesis.

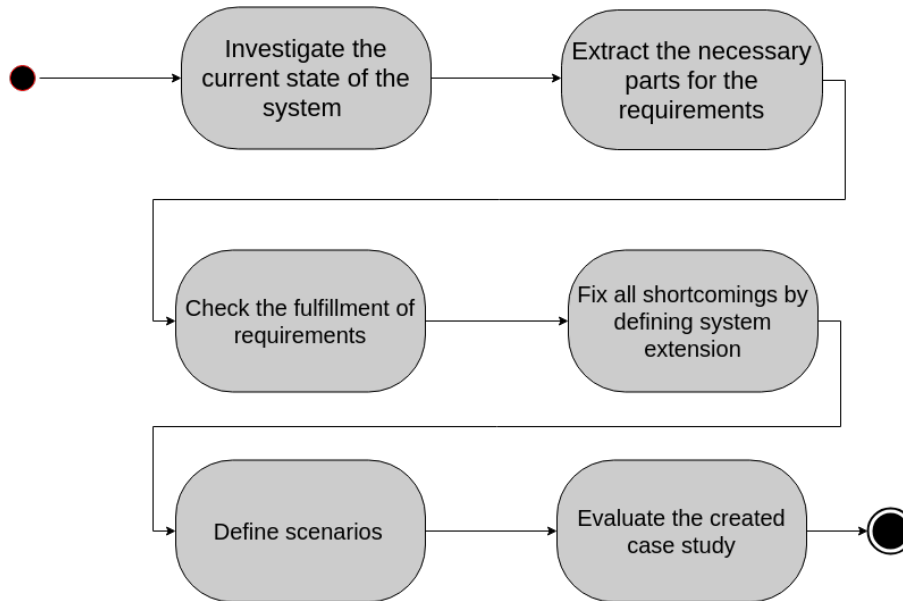
## Foundations

- Security relevant data
  - Data that is worth protecting from threats from the outside
- Data flows
  - Describe the movement of data and the changes to data in a system



Security relevant data: grundlegende begrifflichkeit  
unter der alle zu schützende daten  
zusammengefasst werden  
Data flows mit entity, process und data store mit  
erklärung eines Beispiels

## Procedure overview



Durchgehen um den ganzen Prozess erklären.

## Procedure: requirements for a case study

R1	Component based system
R2	Definition of use case
R3	Security relevant data
R4	Definition of user roles
R5	Defintion of access rights
R6	Definition of the type of data processing

Jede Anforderung einmal genau erklären

R1: Stärke von CBS nutzen

Interfaces, Modularität, einfache erweiterung oder reduktion von einem system

R2: guter Einblick wie das system genutzt werden soll und zum finden von Szenarien

R3: zu schützende daten

R4: rollen die mit dem system interagieren

R5: Zugriffsrechte für CBS (Evered und Bögeholz)

R6: wie werden welche daten in welcher

Komponente verarbeitet. Die Verarbeitung wird durch *Operationen* dargestellt. Jede Operation hat eine oder mehrere Eingaben bzw Ausgaben.

Operationen für die verschiedenen Änderungen eines Datum bzw eines datentyps



## Procedure: shortcomings and milestone

- Check the fulfillment of the requirements for a case study and create a list of shortcomings
- Fix all the shortcomings, for example, adding extensions to the system
  - the milestone is reached, the system is in a state so it is possible to create a case study
- Define different scenarios:
  - Scenarios are used to model typical interactions with the given system
  - From the scenarios the concrete data flows for the system are derived
- Evaluation
  - Quality of access rights
  - Covered information flow classes

**Meilenstein** wurde erreicht wenn alle Anforderungen erfüllt sind

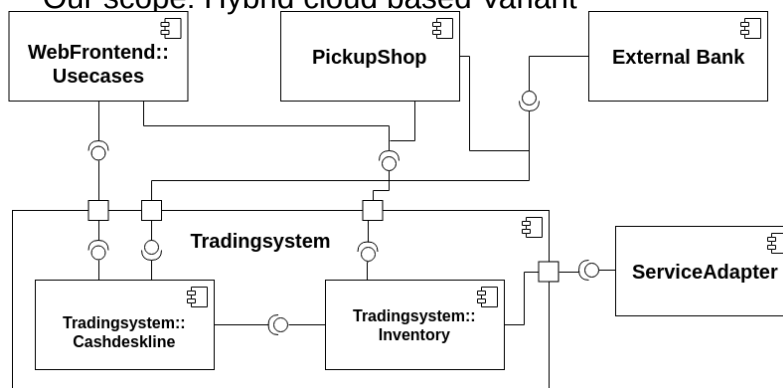
Wenn der Meilenstein erreicht wurde, ist das System oder ein Teil des Systems in einem Zustand in dem es möglich ist eine Fallstudie zu erstellen, da alle benötigten Elemente genau genug definiert sind

**Szenarien** werden definiert für charakteristische/mögliche Interaktionen mit dem System. Daraus werden dann die Datenflüsse generiert. Diese Datenflüsse und die Zugriffsrechte werden dem Modell hinzugefügt werden. Diese Datenflüsse werden dann genutzt um die Sicherheit eines Systems zu validieren.

**Evaluation** : es werden 2 Aspekte evaluiert. Erstens die Qualität der Zugriffsrechte und die abgedeckten Informationsklassen

## CoCoME overview

- CoCoME abstracts the a supermarket system
  - Managing the stock
  - Selling of products
- Our scope: Hybrid cloud based Variant



Motivation ➤ Foundations ➤ Method ➤ Evaluation ➤ Related work ➤ Conclusion

7

9/26/18

Julian Hinrichs

Software-Entwurf und -Qualität  
Institut für Programmstrukturen und Datenorganisation

CoCoME stellt einen großen Supermarkt mit dem normalen Verkauf von Waren und einem Warenlager da

In der hybrid cloud based variant, wurde die Datenbank ausgelagert → Skalierbarkeit

Ein Pickupshop Eingführt der ohne die Kassen zu benutzen Produkte verkaufen kann →

Konkurrenzfähigkeit. Zudem wurden Accounts für alle rollen eingeführt. Mit dem Service adpater wurde der Datenbankzugriff vom Entwickler abstrahiert.

## Fulfillment of requirements R1-R4

- R1: component based system → check
- R2: definition of use cases → check, 13 use cases are defined in various publications
- R3: Security relevant data
  - Account data: security relevant
  - Customer data: security relevant
  - System data: security relevant
  - Product & sales data: security relevant if informations for security relevant classes may be derived.
- R4: user roles
  - 5 different roles defined in the documentation
  - Customer, cashier, StockManager, StoreManager, EnterpriseManager

Motivation ➤ Foundations ➤ Method ➤ Evaluation ➤ Related work ➤ Conclusion

8

9/26/18

Julian Hinrichs

Software-Entwurf und -Qualität  
Institut für Programmstrukturen und Datenorganisation

**R3 : account data** → man kann sich als diese rolle einloggen

**Customer data** → private informationen

list of purchases als neuer datentyp hinzugefügt

**System data** → man kann die anfragen an die DB verändern

**P&S data** → wenn in einem Kompositum mit den anderen Klassen und es eine Informationsgewinnung ermöglicht  
report data type neu hinzugefügt

- R4: user roles

Stock und StoreManager kümmern sich um einen Store (Selling, warehouse)

EnterpriseManager kümmert sich um eine Enterprise, Customer und Cashier sind aufgrund ihrer UC und dem code selbsterklärend

- R5: access rights

- For each user and each component the access rights are defined for each data type
- Four levels of access rights:
  - Level 1: Full Access
  - Level 2: Access to used data
  - Level 3: Access to owned data
  - Level 4 : default/no access

	Webfront-end	TS:inven-tory:app-lication	TS:inven-tory:data
Stock Manager	Customer data : 4 Account data : 3 P & s data : 2 System data : 4	Customer data : 4 Account data : 3 P & s data : 2 System data : 4	Customer data : 4 Account data : 3 P & s data : 2 System data : 4

Wichtige Anforderung, wird aus den bisherigen generiert und kapselt viel entwurfswissen ( Interaktion, aufgaben von rollen, usw)  
Die ACM für den StockManager, nach dem die Mängel behoben wurden  
Anschaulich die vorgeschlagene Definition von Evered und Bögeholz umgesetzt.

## R6: types of dataprocessing

- Define for each component and each datatype the possible operations
- Three types of data processing:
  - Transmission of data
  - Perform relational algebra operations
  - I/O operations

	Customer data	Account data	Product & sales data	System data
Webfrontend	transmit	transmit	I/O operations transmit	Non-existent

Als Beispiel das webfrontend.  
Anschauung, wie die einzelnen Daten innerhalb dieser Komponente verarbeitet werden.

## Definition of scenarios

- Mostly oriented on the use cases or access rights
- The goal is to model interaction with the system
- Scenarios allow to get a detailed insight in the system under investigation
- A desirable goal is to cover all information flows with scenarios to be able to check Non-influence (Non-interference + non-leakage).
- Four information flow classes
  - Illegal information flow
  - Observable information flow
  - Flow from high to low
  - Direct information flow

Szenarien orientieren sich an den use cases da diese interaktionen mit dem system abbilden

Szenarien ermöglichen einen tiefen Einblick in das system. Vorallem wie daten wann und wo verarbeitet werden

Non-interference:

- 1) high inputs beeinflussen keinen low output
- 2) low user können nicht mitbekommen was ein high user macht
- 3) ein observer kann zwischen einzelnen system durchlaufen keinen unterschied feststellen

Non-leakage:

- 1) Verstecken von allen events im system

4 information flow classes

Direct information flow, illegal information flow, flow from high to low, observable information flow

1-3 sind non-interfering zuzuordnen, 4 non-leakage

# Scenario: StockManager requests the report for a customer



## Evaluation: Case Study

Access Rights	Fulfilled ?
Positive	check
Need-to-Know	check
Aspect-oriented	check

Information flow classes	Covered ?
Observable flow	X
Direct information flow	X
Flow from high to low	check
Illegal information flow	check

## Erklären der Checklisten



## Evaluation: Method and PCM extension

- Applicability of the Method
  - Applicable for the examined excerpt of CoCoME
    - method applicable for the investigated and similar systems
- PCM

Relational algebra	check	Change of access rights	check
I/O Operations	check	Alternation of data	check
Transmission of data	check	Access control matrix in the model	X

Anwendbarkeit → durch die Anwendung auf CoCoME gezeigt.

PCM: welche operationen können durch die ADL dargestellt werden

Erklären der Checklisten

## Limitations

- Only one type of system was investigated and from that system only an excerpt.
- We checked three out of seven available criteria due to:
  - Time constraints
  - System constraints
- Not all information flow classes are modeled
  - Two information flow classes are missing
- For each information flow class a violation and a non-violation should be modeled in the case study.

Es wurde nur ein Ausschnitt system untersucht  
Nicht alle sieben Kriterien wurden nicht überprüft  
Nicht alle Information flow classes wurden  
untersucht.  
Nicht für jede Klasse wurde ein positiv und negativ  
beispiel erstellt  
→ **limitiert die Aussagekraft**

## Related Work

- Case studies are commonly used in other fields than computer science, like law, sociology, health care, etc.
- The methodology to create case studies in these fields is well documented and heavily investigated.
- In computer science, case studies are mainly used to investigate how different challenges were tackled.
- A comparable publication: Everend and Bögeholz
  - Evaluation criteria to measure good access rights
  - A shorter version for the creation of a case study, which we used as comparison for our process.

Wenig gefunden, Prozess und Ziel in anderen Fachbereichen außerhalb der Informatik gut und genau dokumentiert

In der Informatik werden Fallstudien mehr dafür verwendet um verschiedene Ansätze zur Lösung eines Problems zu diskutieren

Beispiel: Ein audience response system wie Pingo

Es gab eine Arbeit die aus einem viel kleineren System eine Case study erstellt hat, da wurde nach Möglichkeit verglichen aber unser scope war/ist viel größer. Die Arbeit lieferte Kriterien für gute access rights.

## Future Work

- Method:
  - Apply to other systems (travelsystem, etc)
- Case study
  - Using of the case study in a data based pivacy analysis.
  - Evaluate for the two additional criteria in the short term and all seven criteria in the long term
  - Additional scenarios for the examined excerpt to cover all information flow classes.
  - Define a violation and a non-violation for each information flow class in the scenarios
- PCM
  - Access Control matrix in the same model

## Anwendbarkeit

Auf andere systeme (travelsystem) anwenden

## Case study system

Verwenden um DBPA zu evaluieren

Alle information flow classes abbilden

Für jede information flow classes ein  
positiv/negativ beispiel

## PCM

ACM und OpM(?) in PCM

- Problem
  - Security, especially privacy becomes more important.
  - Ensure compliance by evaluating privacy on an architectural level with a databased privacy analysis.
  - Databased approaches use case studies of a system for validation.
- Idea
  - Introduction of a process to create usable case studies.
- Benefit
  - Usable case studies for data based privacy analysis
- Approach
  - Define requirements for a system so that it is possible to create a case study out of the system .

Schlussfolie, wird so wie ich es verstanden habe nur aufgelegt

# Access control matrix

Beginn von meinen BackupFolien