Verification of Access Control Policies in Software Architectures

~~Proposal~~

Proposal of

Julian Hinrichs

at the Department of Informatics
Institute for Program Structures and Data Organization (IPD)

Reviewer:	Prof. Dr. Ralf H. Reussner
Second reviewer:	Prof. Jun.-Prof. Dr.-Ing. Anne Koziolek
Advisor:	M.Sc. Stephan Seifermann

14. Mai 2018 – 14. September 2018

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

PLACE, DATE

. . . . . .

(Julian Hinrichs)

Abstract

English abstract.

Zusammenfassung

Deutsche Zusammenfassung

Contents

List of Figures

List of Tables

1. Introduction

Nowadays, systems are more complex and connected. Therefore security aspects get more important. By virtue of recent events, users are more aware about their personal data. They want to know how the system they are using processes and stores their data. Therefore, non-functional requirements become more important. General examples for non-functional requirements are law regulations, performance or data transparency. Data transparency in this context means, transparency on how personal data or more common sensible data is processed inside the system.
In my upcoming thesis, I will focus on data protection, a more specific case of law regulations, and data transparency. Both non-functional requirements are well reflected on access control. Therefore, I will identify and analyse dataflows to make sure access control policies aren't violated.
I will work on the architectural level. ~~Furthermore,~~ I will use the Palladio Component Model (PCM) as my architectural descripton language (ADL). Currently, in PCM dataflows aren't a first level entity. So the architecture and dataflows are documented seperately. In the current state some problems arise. One of the main problems is that consistency isn't ensured between the documented dataflows and the design model. Another aspect is that security charateristics often evaluated inside the code. If flaws in the architecture model are detected inside the code, it is expensive to fix them. Expensive in this context means, you have to allocate resources (manpower, time, money, etc) to this task. The problem I try to address it that the early stages of the design phase (in the waterfall model) arent't covered sufficinetly.
As a solution, ~~Stephan~~ Seifermann [3] proposed to extend the PCM-language to include dataflows ~~on the architectural model~~. This allows to have security analyses on the architectural level. I am using PCM, because PCM has more features than only modelling a system. This features include predictions for cost, reliability, maintainbility and performance.
To validate if the addition of dataflows in the architectural model helps to guarentee access control in a system, I will use as a case study the CoCoME system. I chose CoCoME, because it is completly modeled in the PCM. Also an implementation already exists that I can use to verify my claim. Note that the implementation is not complete, so it is likely that I am going to extend CoCoME in my thesis. For now it is sufficient to know that CoCoME is a system which abstracts the inventorymanagement of a big supermarket like Lidl or Aldi.
In CoCoME, I am going to define constraints to ensure non-functional requirements like data protection. After that, I am identifying dataflows inside the system. By using the constraints and the dataflows, I will use logic programming to solve the resulting constraint system. This result, I am going to use and analyse.

2. Basics

In this section, I will give an overview over the basic concepts I am going to use. This includes the CoCoME system, definitions of the basic concepts and th current state of the art.

2.1. Component based systems

Componentbased systems consists of components. A component is defined as unit, which may provide or require interfaces. Big component based system consists out of multiply components, which are connected via interfaces. Each of the components handle a task. Such task could be connecting to a database, displaying the UI or serialize Objects, just to name a few. The different components communicate with each other via predefined Interfaces. The advantage is that component based systems are able to swap the components easily. Therefore, it is possible to add more functionality without a big hassle. Also each component is enclosed by itself and may be developed simultaneously.

2.2. CoCoME

CoCoMe is a such component based System, which abstracts the inventory-management and selling process of a big vendor like Lidl or Aldi. This system is component based (2.1). CoCoME consists of many components, which handle the differnet Use-Cases for CoCoME. The following list is a quick overview of what CoCoMe is (currently) able to achieve.

- Operating of a register cash system
- Processing of inventory and order process of goods
- Providing of an web frontend for the services

CoCoME exist in various variants, which were developed over the time. The codebase, used technologies and the deployment may vary on each variant. Likewise for some variant there are also evolutionary scenarios that may change the deployment of CoCoME or adding additional technologies to the system.For further reading on CoCoME please see the Technical Report [1].
I chose for my case study the Hybrid Cloud Based Variant with the Additon of PickUp-shop. In this variant CoCoME changes from a closed system to an open system. As a closed system CoCoME only handles a finite numbers of users and doesn't processes any sensible data. In the open variant, which I am going to use, CoCoME has an user and store management, in which sensible data is processed.
One of the two main reasons, I picked CoCoME as my case study is that CoCoME is completly modeled in PCM and I can use PCM for obvious reasons. The second reason is simple that an implementation exists.

2.2.1. Archtitectural overview

In this paragraph, I will give a quick architectural overview of CoCoME.

The fundamental layout of the CoCoMe variant, which I am using, is deployed in different layers. On top there is a layer for the frontend, an layer for webservices and a final layer for the program logic. The frontend layer is used to have differnet frontends for customer access. The webservice layer is used to add different services, like analytics, to the system.
The programm logic layer holds the basic logic, the tradingsystem, of CoCoME. This tradingsystem divides into two parts, the cashdeskline and the inventory component. The cashdeskline handles the various user inputs. This inludes the different products an user purchased, the payment method and the displaying actions on the UI. The inventory component handles the connection to the underlying databases and the consitency of the stock items.

2.2.1.1. Adding a PickUp-shop to CoCoME

The concept of a PickUp-shop is, that a customer orders goods online and pay them directly or pay them at the PickUp-shop . The purchased goods are provided by the chosen PickUp-shop, where the customer collects them. This system may be deployed on one or various servers.

2.2.1.2. Database Migration

In this case the CoCoME system deploys the database in a cloud-environment. The purpose of the decison is, if CoCoME grows in user numbers that the scalability of the current cloud isn't sufficient enough anymore and more CPU power is needed. Note that, this change of the cloud environment can be done during the runtime.

2.2.1.3. Adding a Service Adapter

The service adapter is a technique to abstract the database access. This was added to have a layer of abstraction between the database and the application. The benefit of the service adapter is that the database access is handled via an interface and the underlying structure doesn't affect the development process

2.3. Definitions

In this section I will give a definition of some important concepts which are used inside this proposal and the upcoming thesis.

2.3.1. Confidentiality

Confidentiality is present, if there is no unauthorised informatio retrieval possible.

2.3.2. Data protection

Ability of a real person to control the flow of his/her personal data in a given system.

2.3.3. Constraints

Constraints define limits for a system or in this case, a dataflow.

2.4. Palladio Component Model

The PCM allows more than only modelling the systems architecture. It's also possible to analyse the modeled system regarding performance, avaibility . The PCM defines four roles in the development process. These are the component developer, software architect, system deployer and the domain expert. Each role creates and use different models to fulfill their task. In the following, I will give a quick overview over the four roles, mainly which models are created by each specific role. This were all defined in PCM technical report [2].

2.4.1. Component developer

The component developer specigies and implements the component. Furthermore, s/he has to provide additional information, that will b eused for the predictions, PCM is able to compute.

2.4.2. Software architect

The system architect builds the system using components. S/He connects the different

2.4.3. System deployer

2.4.4. Domain expert

2.5. State of the Art

2.6. Concept

In this section I will present the concept of my upcoming thesis. This covers the goal I am trying to achieve, the questions I am going to answer in the process and the methods to verify my results.
The goal of the thesis to analyse data protection and access control in software systems on the architectural level. To achieve that, I am going to answer the following questons:

- How I may model the access control and data protection in an software architecture ?
- How I may verify if access control and data protection are sufficiently covered ?

At last, I will verify my results by having a closer look on the CoCoME system as a case study.

2.6.1. Models and methods

First of all, I will define an access control matrix for the system. This matrix stores which roles may access which data under which circumstances. For example, one can read from the matrix, that the manager is allowed to access the bank account of a company. Policies for data protection ar ealso stored in the access control matrix.
Further, I will extend the component models to include dataflows. This dataflows I am going to use to verify if the policy stated in the access control matrix were violated. For this verification , I define constraints, where dataflows aren't allowed to happen. Then I will define a constraint system by using the constraints and the dataflows. Then I will solve this using logic programming and interpret the result.

2.6.2. Case study: CoCoME

I will now map the plan I described above to CoCoME system. I will add dataflows to the components concerning the PickUp shop and the inventory mamangement. Also I will develop an access control matrix for the CoCoME system.
I'm showing a scenario of how I'll work with the models and methods described above.

2.7. Scenario

The scenario I am going to use, is the Use Case 13 (UC13) specified in Technical Report [1] . In this use case states that the stockmanager is able to retrieve specific data concerning a specific customer. This specific data are the purchased products by the customer.
Currently the Technical Report [1] only states, the ID of a customer is given to an authenticated stockmanager. Then the stockmanager is able to retrieve all the purchased products of the specific customer, assimilated to a report. For this scenario the resulting dataflow is shown in Figure 2.1.

In this dataflow, one can see that two different queries to the databases are made. Also alot of personal information is going to be extracted.
The flaws in this scenario are, that the stockmanager is able to retrieve information about a specific customer, which he doesn't need to do his work. The stockmanager should only which products are purchased. Also , it is possible to identify the customer in the system. The ID, which is known to the stockmanager allows to get the customer object in the system. This object contains first name, last name, email address and the preffered store. This is a data protection breach in the system. Further, the principle of collecting as little as possible data is violated.

2.8. Evaluation

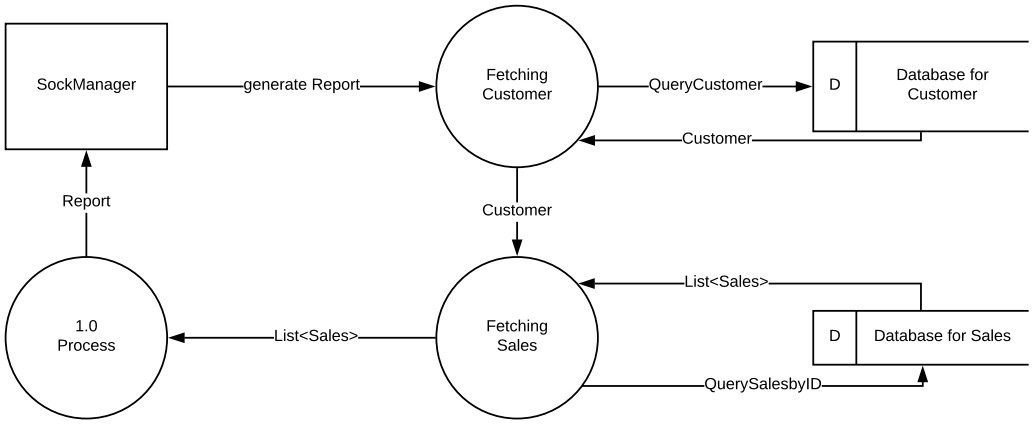

Figure 2.1.: Dataflow for the implementation using a list in the customer object

3. Organisation

3.1. Shedule

The next figure shows a rough overview over my schedule in the thesis.

3.2. Riskmanagement

In this section I will focus on risk management of my thesis. Therefore, I will point out some aspects, that may cause delay of my work.

- **Roles in CoCoME**
  Currently a rolemanagement is not implemented in CoCoME. I will add this. For this I may use more time than I expected.
- **Implementation of Scenarios**
  Some of the scenarios I am going to use, aren't par tof CoCoME yet. It may take more time than expected to implement these.
- **Extension of PCM**
  Currently PCM doesn't support dataflows in the architecture model. It may cost more time than anticipated to extend the PCM.

Bibliography

[1] Robert Heinrich, Kiana Rostami, and Ralf Reussner. “The CoCoME Platform for Collaborative Empirical Research on Information System Evolution”. In: (2016).

[2] Ralf Reussner et al. *The Palladio Component Model*. Tech. rep. 14. 2011. 193 pp.

[3] Stephan Seifermann. “Architectural Data Flow Analysis”. In: *13th Working IEEE/IFIP Conference on Software Architecture, WICSA 2016, Venice, Italy, April 5-8, 2016*. 2016, pp. 270–271. DOI: 10.1109/WICSA.2016.49. URL: https://doi.org/10.1109/WICSA.2016.49.

A. Appendix

A.1. First Appendix Section

Figure A.1.: A figure

...