



# A Mobility Case Study Framework for Validating Uncertainty Impact Analyses regarding Confidentiality

Bachelor's Thesis of

Denis Priss

at the Department of Informatics
KASTEL — Institute of Information Security and Dependability

| | |
|---|---|
| Reviewer: | Prof. Dr. Ralf Reussner |
| Second reviewer: | Prof. Dr.-Ing. Anne Koziolek |
| Advisor: | M.Sc. Sebastian Hahner |
| Second advisor: | M.Sc. Maximilian Walter |

21. June 2022 – 21. October 2022

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

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.

**Karlsruhe, 21.10.2022**

. . . . . .

(Denis Priss)

# Abstract

Software systems have become more global, interconnected, and complex in the last two decades. Confidentiality has become an important security requirement for information systems. The later confidentiality breaches are discovered during the development process, the more expensive it can be to fix them. In the early stages of the design, confidentiality breaches might arise from uncertainties about the system and its environment. Therefore, it is essential to consider such uncertainties already when designing the system and assess their impact on confidentiality. Approaches exist that are intended to support software architects in examining the effects of uncertainty on confidentiality. However, these have not yet been extensively evaluated and their benefit has not yet been independently proven. In order to increase the validity, further independent case studies should be carried out. A uniform procedure is necessary to generate consistent results. While there is work in this area in general, it is not specific enough to meet the requirement.

This thesis aims to fill this gap by introducing a framework that includes an investigation process and a case study protocol. The investigation process consists of general steps necessary for case studies and steps specific to this analysis class. The case study protocol includes requirements for the case study and for the case study design, which builds on existing literature. This framework should help researchers to carry out further case studies to validate the *Uncertainty Impact Analysis (UIA)* in a structured manner and thus also to explore uncertainties and their impact on confidentiality. In order to evaluate the framework, a mobility case study is conducted that independently evaluates a *UIA*. This mobility case study is a further contribution to this thesis.

Finally, this thesis evaluates the quality of the investigation process and the case study, as well as the completeness of the aspects to be investigated. The evaluation plan and the evaluation design are based on the defined requirements. The purpose of the evaluation is to examine whether the investigation process and the case study strategy have a sound basis, are feasible, and promote meaningful results. The requirements coverage of the aspects to be examined reaches about 80%. The introduced framework supports the validation of the *UIA*: Results obtained in the case study imply that the structural propagation of uncertainties alone is insufficient to provide software architects with a reliable starting point for further confidentiality analyses.

# Zusammenfassung

In den letzten zwei Jahrzehnten sind Software-Systeme globaler, vernetzter und komplexer geworden. Dabei ist die Vertraulichkeit zu einer wichtigen Sicherheitsanforderungen an Informationssysteme geworden. Je später Vertraulichkeitsverletzungen im Entwicklungsprozess festgestellt werden, desto kostenintensiver kann deren Behebung werden. Bereits im frühen Entwurf existieren Ungewissheiten, sowohl über das System als auch dessen Umgebung, die sich auf die Vertraulichkeit auswirken können. Daher ist es wichtig bereits zur Entwurfszeit solche Ungewissheiten zu berücksichtigen und deren Auswirkungen auf die Vertraulichkeit einzuschätzen. Es existieren Ansätze, die Softwarearchitekten und Softwarearchitektinnen unterstützen sollen, Ungewissheitsauswirkungen auf die Vertraulichkeit zu untersuchen. Diese wurden jedoch noch nicht umfangreich evaluiert und deren Nutzen noch nicht unabhängig nachgewiesen. Um die Validität zu erhöhen, sollten weitere unabhängige Fallstudien durchgeführt werden. Dabei ist ein einheitliches Vorgehen notwendig, um konsistente Ergebnissen au erzeugen. Obwohl es allgemein Arbeiten in diesem Bereich gibt, sind diese nicht spezifisch genug um die Anforderung zu erfüllen.

Diese Arbeit soll diese Lücke schließen, indem ein Rahmenwerk vorgestellt wird, welches einen Untersuchungsprozess und ein Fallstudienprotokoll beinhaltet. Der Untersuchungsprozess besteht aus allgemeingültigen Schritten, die für Fallstudien notwendig sind und Schritten, die speziell für diese Klasse von Analysen notwendig sind. Das Fallstudienprotokoll beinhaltet Anforderungen an die Fallstudie und an den Fallstudienentwurf, der auf vorhandener Literatur aufbaut. Dieses Rahmenwerk soll Forschenden helfen, weitere Fallstudien zur Validierung der Ungewissheits-Auswirkungs-Analysen strukturiert durchzuführen und damit auch Ungewissheiten und deren Auswirkung auf Vertraulichkeit zu erforschen. Um das Rahmenwerk zu evaluieren, wird eine Mobilitätsfallstudie durchgeführt, mit der eine Ungewissheiten-Auswirkungs-Analyse unabhängig evaluiert wird. Diese Mobilitätsfallstudie stellt einen weiteren Beitrag dieser Arbeit dar.

Abschließend wird in dieser Arbeit die Qualität des Untersuchungsprozesses und der Fallstudie sowie die Vollständigkeit der zu untersuchenden Aspekte evaluiert. Dabei richten sich der Evaluationsplan und der Evaluationsentwurf an den definierten Anforderungen aus. Durch die Evaluation soll untersucht werden, ob der Untersuchungsprozess und die Fallstudienstrategie eine fundierte Basis besitzen, durchführbar sind und aussagekräftige Ergebnisse fördern. Dabei wird eine Anforderungsabdeckung an die zu untersuchenden Aspekte von rund 80% erreicht. Das vorgestellte Rahmenwerk unterstützt die Validierung der Ungewissheiten-Auswirkungs-Analyse: Im Rahmen der Fallstudie gewonnene Ergebnisse lassen darauf schließen, dass die strukturelle Ausbreitung von Ungewissheiten alleine nicht ausreichend ist, um den Softwarearchitektinnen und Softwarearchitekteneinen verlässlichen Ausgangspunkt für die weitere Vertraulichkeitsanalysen zu geben.

# Contents

# List of Figures

# List of Tables

# 1. Introduction

Software systems have become more global, interconnected, and complex in the last two decades [23]. Confidentiality has become an important security requirement for information systems. According to International Organization for Standardization, confidentiality is "property that information is not made available or disclosed to unauthorized individuals, entities, or processes" [11]. The later confidentiality breaches are discovered during the development process, the more expensive it can be to fix them [7]. In the early stages of the design, confidentiality breaches might arise from uncertainties about the system and its environment. Therefore, it is essential to consider such uncertainties when designing the system and assess their impact on confidentiality.

Researchers are developing solutions that support software architects analyzing uncertainties and assessing their impact on confidentiality, such as the Uncertainty Impact Analysis (UIA) proposed by Benkler [4]. However, these have not yet been extensively evaluated, and their benefit has not yet been independently proven. In order to increase the validity, further independent case studies should be carried out. A uniform procedure is necessary to generate consistent results and guide the researcher through the investigation.

The uncertainty propagation is a "contemporary software engineering phenomenon within its real-life context" [23], making it an excellent research strategy to validate the UIA. While there is work in this area in general, e.g., guidelines and examples on case study research in software engineering by Runeson et al. [23] or a more recent work on the same topic by Wohlin [26], these contributions are not specific enough to our requirements. More specific contributions do not introduce suitable procedures for our needs.

To close the gap, we introduce a framework that consists of an investigation process and a case study protocol for mobility case studies that validate a UIA regarding confidentiality. The investigation process combines general steps necessary for case studies [23] and steps specific to this analysis class [4]. The process is iterative and incremental, which enables improving quality over iterations and conducting a case study of high complexity. The case study protocol follows Runeson et al.'s [23] and Brereton et al.'s [8] guidelines. It includes requirements to ensure the study's quality and comprehensiveness and a case study design that specifies the its characteristics and data collection and analysis methods. This framework should help researchers to carry out further case studies to validate the UIA in a structured manner and thus also to explore uncertainties and their impact on confidentiality.

We evaluate the framework by conducting a mobility case study that validates a *UIA* regarding confidentiality. We apply the Goal-Question-Metric (GQM) approach [2], considering the defined rationals, objectives, and requirements. The evaluation aims to examine whether the investigation process and the case study strategy have a sound basis, are feasible, and promote meaningful results. The requirements coverage of the aspects to be examined reaches about 80%. The introduced framework supports the validation of the *UIA*:

Results obtained in the case study imply that the structural propagation of uncertainties alone is insufficient to provide software architects with a reliable starting point for further confidentiality analyses.

The core contribution of this thesis is a framework consisting of (1) a set of requirements concerning the case study's quality and comprehensiveness, (2) an investigation process for case study conduction, and (3) a case study protocol that supports researchers during the investigation. To be precise: The case study is a mobility case study for validating an Uncertainty Impact Analysis regarding confidentiality. The second contribution is a mobility case study conducted according to our framework.

The remainder of this thesis is structured as follows: We start by providing foundations for our approach in Chapter 2. Chapter 3 summarizes state of the art and outlines the gap we aim to fill. We introduce our framework, the investigation process, and the case study design in Chapters 4 and 5, respectively. We introduce our framework the investigation process and the case study design in Chapters 4 and 5 respectively. In Chapter 6, we conduct a mobility case study according to the proposed framework. We use this case study to evaluate our framework in Chapter 7. Finally, we summarize our findings and provide a short outlook on future work in Chapter 8.

# 2. Foundations

This chapter gives foundations on the topic required for this thesis. An obstacle to a case study's success is that case studies are mostly conducted by biased researchers [23]. Runeson et al. [23] state that the bias can be mitigated "by applying proper research methodology" [23, p. 4]. Section 2.1 briefly overviews the general case study and the general case study creation methodology. For the case study we conduct, the foundations of the mobility domain are essential. Section 2.2 provides the required foundations for the domain. We conclude this chapter by explaining Palladio in Section 2.3 , and after defining uncertainties in Section 2.4.1 , we present the Palladio add-on Uncertainty Impact Analysis (UIA). in Section 2.4.2.

## 2.1. Case Study

As the thesis title reveals, we conduct a case study to validate the UIA. In this section, we first clarify what a case study is and its characteristics, then present a simplified process of how case studies are conducted.

> "Case study in software engineering is an empirical enquiry that draws on multiple sources of evidence to investigate one instance (or a small number of instances) of a contemporary software engineering phenomenon within its real-life context, especially when the boundary between phenomenon and context cannot be clearly specified." Runeson et al. [23, p. 12]

In other words, a case study is a research approach to investigate phenomena within a real-life setting, where it is sufficient to examine "a single case or a small number of cases" [24]. The goal of the investigation can be to *observe* (descriptive), *explain* (explanatory), or *explore* (exploratory) phenomena or to *evaluate* research (evaluatory) [24].

The case study is conducted iteratively and incrementally [23]. The last property implies that the further iterations build upon data collection, analysis, and experience from previous iterations. This characteristic enables researchers to adjust the case study to their needs and findings. This flexible approach has some drawbacks [23], e.g., a replication of a case study can produce different distinct results even if the investigators try to apply the case study as it was done originally.

The following list presents the five identified process steps, and the paragraph below explains these steps in more detail.

1. Case Study Design
2. Preparation for data selection
3. Collecting evidence
4. Analysis of collected data
5. Reporting

During the case study design phase, the objects, goals, and questions are clarified , and the conduction is planned [23]. The design shall not be restrictive, as the new knowledge gained from data analysis may generate new or other perspectives to explore. During the preparation for data selection, investigators clarify data to collect and select suitable methods for data collection. The planned activities are now executed , and the evidence is collected. At this moment, the collected data is raw and must be systematically organized to ease pattern recognition to "understand what actually has happend in the studied case" [23, p. 61]. The final stage of the iteration is reporting, which is critical as stakeholders may have different levels of knowledge and may be interested in different aspects of the case study. Due to the iterative character, the steps might be executed multiple times.

## 2.2. Mobility Data Specification

In recent years, many novel modes of transport have come into existence, e.g., e-scooters and bike shares [13]. Since the use of the new transport possibilities is still relatively new and therefore unknown, it is unclear whether the existing infrastructure is suitable for such usage. On the other hand, these novel modes of transport require clear regulations that provide a solid ground for all stakeholders. Usage data could help improve the infrastructure, and therefore the users would benefit from it. Data can also help establish suitable policies. The mobility service providers possess such data and can provide it to authorities. This is the point where Mobility Data Specification (MDS) intervenes to improve communication between mobility service providers and authorities.

MDS is a free and open-source standard that brings several advantages. It "standardizes communication and data-sharing between cities and private mobility providers" [13]. So, MDS improves collaboration between cities and providers. Cities can adjust their policies nearly in real-time, while providers gain more certainty because the policy has fewer gray areas. Through this standard, providers enable access to specific near-real-time and historical data, which authorities can use for better decision-making on infrastructure and policies. The cities can also provide the data to concerned citizens. [13]

MDS, at its core, is a set of six Application Programming Interfaces (APIs): (1) provider, (2) agency, (3) policy, (4) geography, (5) jurisdiction, and (6) metrics [13]. Each API contains the endpoints specification described by *JSON Schema* [14]. This schema defines a *RESTful API* so that the response JSON document is valid against this schema [13, 14]. For our purpose, the essential APIs are (1)-(3).

**Provider API** [14]: Is implemented by the provider and consumed by the agencies. This API enables access to near-real-time and historical information. MDS classifies the information on trips that customers have covered, reports (revealed by the provider), and vehicle status changes as historical data. The vehicle status, recent events, and stops are available as near-real-time data.

**Agency API** [14]: Is "intended to be implemented by regulatory agencies and consumed by mobility providers" [14]. Open Mobility Foundation (OMF) considers that vehicles can switch the service areas of cities. This API enables registering and deregistering the vehicles when entering or leaving a service area. Furthermore, this API allows mobility providers to update information about one or multiple vehicles.

**Policy API** [14]: Regulatory agencies implement this API, allowing them to adjust municipal policies in nearly in real-time. Policy rules can be speed limits, fees, and many others, which shall apply in certain areas and for specific time intervals. Additionally, agencies can notify vehicle users about various issues through this API.

## 2.3. Palladio

Refinements of architectural requirements can lead to high costs for adjusting existing software [7]. This makes it all the more critical for larger projects to establish a qualitative software architecture as a solid base for the actual software implementation.

Palladio provides an approach to this. It consists of three parts: (1) Palladio Component Model (PCM), (2) analysis techniques based on the PCM, and (3) a development process for developing component-based software systems [21, p. 11]. Shortly stated, the Palladio approach follows the Component-Based Software Engineering (CBSE) paradigm. In CBSE, the software architecture comprises multiple interconnected building blocks encapsulating certain functionality. PCM is a meta-model that supports software architects modeling component-based architectures with a focus on quality predictions [21, 3]. The focus on quality predictions means that PCM enables appending additional information, quality descriptions, and quality annotations about the model and the execution environment. The analytical techniques use this additional information to analyze the model automatically and help stakeholders with decision-making. The Palladio development process introduces different roles for development. Each role contributes a unique set of information to the model, e.g., the system deployer specifies the available processing resources. [20]

In the early stages of Palladio, its main purpose was performance analysis. Since then, additional analysis add-ons have joined the Palladio ecosystem, such as the UIA [4, p. 27].

## 2.4. Uncertainties in Software Architecture

In this section, we first define uncertainties before introducing the UIA proposed by Benkler [4].

### 2.4.1. Uncertainty Definition

Walker et al. define uncertainties as “ any deviation from the unachievable ideal of completely deterministic knowledge of the relevant system ” [25]. To classify uncertainties, they introduce three dimensions: *location, level, and nature*. The location defines where the uncertainty emerges in the model. The level of uncertainty ranges from *total ignorance* to *complete deterministic understanding* [25]. For certain uncertainty classes, the level of uncertainty may be improved by additional research. If an uncertainty belongs in such a class can be derived from the nature of the uncertainty [25].

### 2.4.2. Uncertainty Impact Analysis

As stated in the introduction, unconsidered uncertainties during the design time can cause difficulties in fixing emerged confidentiality breaches. An approach that should support architects in assessing the impact of uncertainties regarding confidentiality is Benkler's UIA [4].

The work by Benkler [4] has two main outcomes. First, it provides an *uncertainty template* "that enables software architects to structurally derive types of uncertainties and their impact on architectural element types for a domain of interest." [4]. To elicit a more detailed model, a more experienced architect, also called an *expert architect*, is better suited to work with the uncertainty template.

An architect with less experience, also called a *user architect*, uses the uncertainty impact analysis, which is the UIA Add-on for the Palladio project. The UIA "enables software architects to specify which architectural elements are directly affected by uncertainties" [4]. In other words, the user architect annotates the existing software model with previously elicited uncertainty types by expert architects.

Now, the UIA automatically propagates the uncertainties through the annotated software model and derives "further architectural elements which are potentially affected" [4] by uncertainty.

# 3. State of the Art

This chapter presents the state of the art and highlights deficiencies from the perspective of our contribution. These deficiencies represent a gap in the current state of the art, which we fill in the following sections. Section 3.1 presents case studies that deal with uncertainties and their impact on confidentiality. To the best of our knowledge, there is only one case study that fits in this section. We list case studies regarding quality predictions of component-based software architectures in Section 3.2. Section 3.3 presents elaborations on frameworks for case studies.

## 3.1. Case Studies for Validation of Uncertainty Impact Analyses regarding Confidentiality

Benkler [4] not only proposed the uncertainty template and the Uncertainty Impact Analysis (UIA), which we described in Section 2.4.2, but he also conducted a case study to evaluate his approach [4]. However, introducing a protocol for case study conduction on how to conduct a case study for validating a UIA approach was not within the scope of his work.

The case study's domain is a cross domain between *Digital Contact Tracing* and *Healthcare*. The *Corona Warn App* served as a case since it has high confidentiality requirements, it is open source, and the design follows the component-based approach. Benkler [4] consulted the freely available design documents, a detailed review of these design documents, and an already performed and related case study as sources of information [4].

Benkler [4] divided the evaluation of the UIA into two parts. First, the evaluation of the uncertainty template, and second the evaluation of the UIA itself. He evaluated according to the Goal-Question-Metric (GQM)-Plan introduced by Basili, Caldiera, and Rombach [1]. Basili, Caldiera, and Rombach [1] argue that this approach ensures measuring "in a purposeful way" [1].

The full description and the data collection on the evaluation with goals, questions, and the used metrics can be found in the master thesis [4] and on Zenodo [5].

Benkler [4] reasons that the results were promising. However, he also determined some weaknesses and proposed conducting additional case studies with other cases and in other domains [4, p. 127]. Nevertheless, this requires a well-defined case study creation process, which we introduce in this thesis. The next chapter provides a draft for such process.

## 3.2. Case Studies with Palladio

Common Component Modeling Example (CoCoME) is a project by several researchers to compare and classify various component modeling approaches [10]. These approaches were compared using a common component modeling example, as CoCoME stands for. A *Trading System* served as a common example.

Palladio is one of the component modeling approaches used on the CoCoME to showcase the approach. Krogmann and Reussner [12] modeled the CoCoME and conducted a performance study on this case. They presented a component-based software development process; however, they did not investigate the uncertainty impact regarding confidentiality. The presented setting is rather a use case but not a case study.

## 3.3. Case Study Frameworks

We could find several contributions proposing a framework or at least guidelines and design specifications.

In Chapter 2, we briefly presented Runeson et al.'s guidelines on case study research in software engineering. The authors proposed a case study investigation process. And extensive guidelines what investigators must consider during the case study conduction. In particular, the Runeson et al. draw researchers attention to case study design [23]. They outline that the case study design can be supported by Brereton et al.'s protocol template [8].

Brereton et al. conducted several case studies to investigated the systematic literature reviews. They developed a case study protocol to ensure consistency over the case study series. They aligned their template to Yin's approach because their goals had consensus with those of Yin.

The case study protocol template consists of 11 sections:

1. Background
2. Design
3. Case Selection
4. Case Study Procedures and Roles
5. Data collection
6. Analysis
7. Plan Validity
8. Study Limitations
9. Reporting
10. Schedule
11. Appendices

The elaboration by Runeson et al. presented a case study conduction process [23]. Unfortunately, this process is very broad and does not consider uncertainty-related steps. This also applies to the protocol template [8]. However, these elaborations are useful for us to build our framework upon them.

# 4. Investigation Process

This chapter introduces a case study conduction process. Figure 4.1 illustrates this iterative process. We believe this method is suitable because it combines (1) the five steps identified by Runeson et al. [23], (2) the elements of the case study design identified by Runeson et al. [23], and (3) the work by Benkler [4]. Respectively, Section 2.1 and Section 2.4.2 present these foundations. The presented process induces "procedures governing field procedures" [8] and required roles.

## 4.1. Roles

Researchers can bring different strengths and also different weaknesses related to the topic of the case study. Therefore, defining and assigning roles to the respective researchers according to their strengths makes sense. Additionally, roles increase transparency because the areas of responsibility are clear to every researcher. In this section, we present four roles, which we assign to the corresponding process steps in the next section.

**Investigator** All participants in the case study investigation are investigators. This role manages the organizational parts of the case study conduction. Investigators work with stakeholders to define requirements for the case study and design the case study by following Runeson et al.'s guidelines [23]. The reviewer role is not explicitly defined; however, every investigator can review case-specific aspects, e.g., decisions during the study design.

**Expert Architect** This role is derived from Benkler's Uncertainty Impact Analysis (UIA) approach [4]. According to Benkler, the *Expert Architect* is an experienced software architect, whereas the *User Architect* is less experienced. *Expert Architects* gather information on the domain of the case study, use their experience to extract Architectural Design Decisions (ADDs), and instantiate the uncertainty templates. For our process, they must also be able to perform uncertainty impact analysis manually and document their findings.

**User Architect** This role is derived from Benkler's UIA approach [4]. The *User Architects* require less experience than the *Expert Architects* [4]. They use an uncertainty template to annotate the model and start the UIA. By annotate, we mean assign uncertainties to specific architectural elements in the model. Furthermore, for our process, *User Architects* can create a component-based software architecture from specifications.

**Analyst** This role is case study-specific. *Analysts* consume findings from *Expert* and *User Architects*, evaluate these data and report the findings to stakeholders.

## 4.2. Procedures

The flowchart in Figure 4.1 presents the required steps for the case study conduction and evaluation of this case study. Each step is assigned to a specific role, described in Section 4.1. We apply this process during the entire thesis. We designed this process to be iterative and incremental, as proposed by Runeson et al. [23].

**/1/ Case Study Design** In this first step of the iteration, the investigators define the study's rationales, objectives, and requirements. Defining these first keeps the investigator from losing focus and helps concentrating on the essentials [23]. The requirements are intended to ensure a good quality case study. In particular, requirements define aspects to be examined by the case study. To measure the progress and quality of the case study, investigators elicit goals, questions, and metrics to the Goal-Question-Metric (GQM)-Plan [1]. Further helpful aspects to look at during design are the study classification, the units of analysis, and case selection. Runeson et al. point out that to prevent the case study from failing because of legal, ethical, and professional issues, investigators must consider these requirements already in the early stages of the case study [23].

In further iterations, investigators might adjust the design set to reflect the new knowledge.

**/2/ Preparation for Data Collection** Before beginning to collect data, investigators must determine what data is needed and define data collection methods to determine the required data for later analysis. This procedure prevents unnecessary data from being collected or missing necessary data. Another starting point for data collection is the model. For this, a case must be worked out that considers all requirements. The case is drafted relatively early but can be adjusted at any time during the case study conduction. If the case is changed, steps **/4/** to **/7/** may also need to be refined. Finally, investigators must clarify where and how the data is stored, managed, and accessed.

**/3/ Deriving Uncertainty Types** *Expert Architects* perform this step because of their experience, as described previously. This step is mandatory when conducting a case study in a new domain. *Expert Architects* systematically review the literature about the domain and apply the uncertainty-type derivation approach proposed by Benkler [4]. According to this approach, researchers first derive ADDs and then, based on these, derive uncertainty types that may exist in this domain. The result of this procedure is an uncertainty template that is required for the next steps, **/5/** and **/6/**.

**/4/ Create / Extend Model** *User Architects* use the defined case and create a component-based architectural model. For that, they use the Eclipse IDE (Eclipse Modeling Project) and Palladio [21] because the UIA is the main subject of this case study, and it is a Palladio add-on [4]. They might extend this model in future iterations to increase complexity and to reflect the new knowledge.

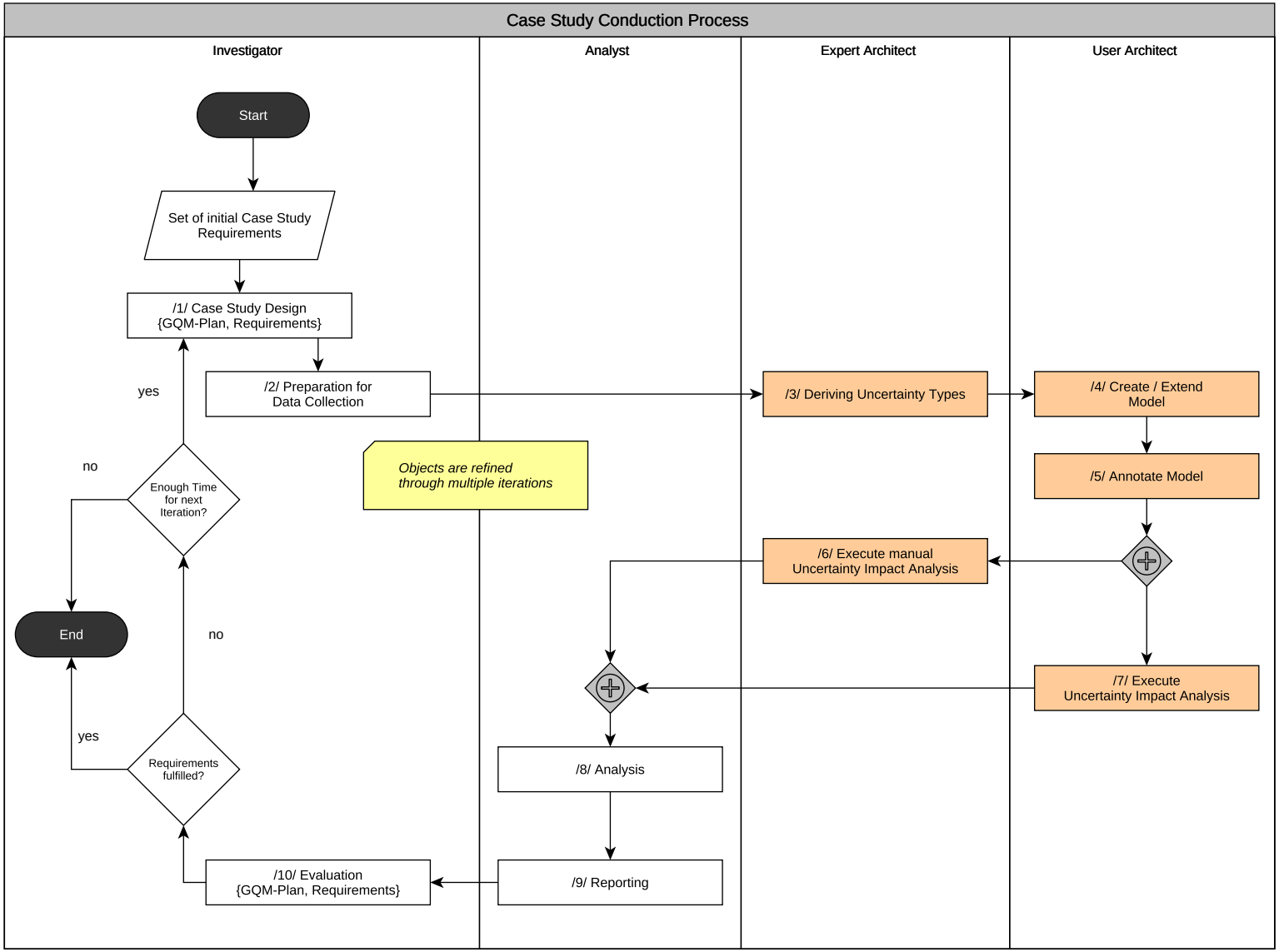


Figure 4.1.: Flowchart of the case study conduction process.

**/5/ Annotate Model** First, *User Architects* must define scenarios. These scenarios describe which uncertainty types directly impact which architectural elements. The *Expert Architects* use these descriptions to perform step **/6/**.

Second, *User Architects* use these defined scenarios to instantiate uncertainty models proposed by Benkler [4]. This process is called *uncertainty annotation*. Uncertainty models are necessary for step **/7/** because the UIA uses them to execute propagation algorithms.

**/6/ Execute manual Uncertainty Impact Analysis** *Expert Architects* use uncertainties from a defined scenario and the architectural model created in step **/4/** for manual propagation and to examine which architectural elements are possibly affected by these uncertainties. We mean manual propagation, the assessment of uncertainties and how they potentially propagate through the architecture, while any UIA does not support the assessment. This is the essential difference to the *User Architects*' approach in steps **/6/** and **/7/**.

**/7/ Execute Uncertainty Impact Analysis** *User Architects* use the uncertainty models created in step **/5/** to prepare the UIA for the analysis and start the automated uncertainty propagation analysis. The UIA outputs the potentially affected architectural elements and the related propagation paths. *User Architects* document these results for the analysis step.

**/8/ Analysis** The *Analysts* use the uncertainty template, scenarios, results from the manual propagation, and results created by the UIA to execute previously defined data analysis procedures. Finally, they document their findings.

**/9/ Reporting** The *Analysts* assess the report's setting, e.g., the addressees of the report, their level of knowledge, and their concerns. The *Analysts* select the subset of required information according to the addressee's demands. Then, they gather the required artifacts and documentation to enrich these findings and compose an accurate report.

**/10/ Evaluation** This step deals with the quality and comprehensiveness of the case study. In step **/1/** a GQM plan was designed. *Investigators* use this plan to evaluate if this case study iteration meets the requirements. The evaluation provides information on whether improvements should be implemented in the next iteration. Furthermore, *Investigators* use the results to decide whether further iterations should be performed.

**Case Study Determination** Generally, the process determines if the case study fails. However, there are other indicators of when to stop the investigators. The process determines if resources have been exhausted, e.g., due to time constraints. Successful process completion occurs when all requirements are met because requirements specify what should be researched and the required quality of the results.

# 5. Case Study Design

In this chapter, we design a mobility case study for validating an Uncertainty Impact Analysis (UIA) regarding confidentiality. Chapter 6 presents a conducted mobility case study that follows this design. We follow Runeson et al.'s [23] guidelines on case study design to create a sophisticated case study design. We evaluate the quality of this design in Chapter 7 after each iteration. The case study design is part of the proposed investigation process in Chapter 4.

We start this chapter by providing background information on why we want to conduct a mobility case study in Section 5.1. We define the rationales and objectives of the study in Section 5.2. In Section 5.3, we define requirements to ensure that the case study has good quality and does not miss out on essential aspects to be investigated. We derive research questions from the defined rationales, objectives, and requirements. Section 5.4 presents these research questions. Afterward, we classify the required study in Section 5.5 and define case selection criteria that must be considered during the case study conduction. Section 5.7 is the most crucial for the case study investigation. In this section, we first provide essential definitions used throughout the entire section. Subsequently, we define for each research question how the data should be collected and how to analyze the data to answer the research questions. Finally, to prevent our study from failing because of legal, ethical, and professional concerns, we elaborate on these in Section 5.8.

Please note: Some sections are described in more detail in Chapter 7 in order to reduce redundancies. We explain threats to validity and how we handled them in Section 7.3. Assumptions and limitations are presented in Section 7.4. Information on data storage and availability is described in Section 7.5.

## 5.1. Background

During software project design, architects must deal with uncertainties. These uncertainties might lead to severe problems if they are not considered early enough during the design phase. Considering uncertainties during design means the architects manually assess the effect on the confidentiality of architectural elements caused by uncertainties. This manual uncertainty propagation and impact assessment can cause much effort.

Benkler proposed a Palladio add-on that automatically analyses the structural propagation of uncertainties regarding confidentiality through software architecture. He claims that this add-on can reduce the assessment effort. The designation of this add-on is UIA. He conducted a case study within the healthcare domain to validate his contribution (UIA) and received promising results.

Benkler evaluated his contribution, which is a potential threat to validity. Our case study evaluates the UIA further but in the mobility domain. Additionally, we propose a

case study protocol that should guide future researchers who would like to re-evaluate the UIA or extend the body of knowledge on the mobility domain regarding uncertainties and their impact.

## 5.2. Rationale and Objectives of the Study

As mentioned in the previous section, two pursued rationales exist behind this case study. First, we want to re-evaluate the UIA proposed by Benkler in the mobility domain [4]. The second rationale arises from the opportunity that we are already in the mobility domain and investigate the uncertainties. Therefore, the second rationale is investigating how uncertainties propagate in the mobility domain. However, for this rationale, we assume that Benkler's uncertainty-type derivation approach is correct [4]. This assumption is discussed in Section 7.4 in more detail.

The objectives, also known as purposes, follow the rationale [23]. Thus, the numbering of objectives corresponds to the numbering of the rationales. We expect to confirm Benkler‘s results regarding the UIA. Should some of the results not be confirmed, we expect to find corresponding weaknesses. The second objective is to find uncertainty-related aspects specific to the domain of such aspects, similar to those found by Benkler [4]. Benkler presumes that the uncertainty template he derived from the healthcare domain could fit other domains. This can be understood as a sub-objective of the second objective, which is to confirm or disprove Benkler‘s supposition.

## 5.3. Requirements

This section specifies the requirements for the case study. These requirements are divided into two groups. The first group contains requirements that ensure the quality of the case study. These requirements are derived from the characteristics of a *good case study* proposed by Runeson et al. [23].

The second group defines points of interest the case study must investigate. These requirements align with the case study's objectives discussed in Section 5.2. Benkler's master's thesis [4] provided additional uncertainty-specific knowledge.

The requirement specifications follow a pattern: (1) the requirement is *explained*, (2) we provide *rationales* for why we defined it. The requirements ID comprises a prefix R for **R**ational, followed by a unique number.

### 5.3.1. Case Study-Specific Requirements

This section presents case study-specific requirements **R1** to **R6**. These requirements concern the quality of the case study.

**R1** Embedded in a real-world context: The case study investigates a phenomenon within its real-life context [23]. The goal of the UIA is to help software architects with uncertainty propagation and assess the impact on confidentiality. This is a problem

that exists in a real-world context. Thus, to evaluate the UIA the case must be embedded in a real-world context (not only in an academic environment).

**R2** Feasibility: Runeson et al. [23, p. 19] list three criteria that characterize a feasible case study: (1) contemporaneousness, (2) type of research questions, and (3) the degree of control is less critical. Researchers must be able to conduct the planned case study; otherwise, it is useless.

**R3** Theoretical Basis: According to Runeson et al. [23], a good case study must add to "existing knowledge by being based on previously established theory, if such theory exists, or by building a theory." The case study must consider a wide range of available information to provide a solid basis for the research.

**R4** Comprehensibility: The case study provides extensive information, e.g., about which, why, and when decisions were made. Investigators must achieve comprehensibility in all case study-related documents: case study design, data collection, and analysis. Runeson et al. highlight the importance of a clear chain of evidence [23]. These provisions help during the re-evaluation and replication of the case study; on the other hand, they help during report creation to ensure the stakeholders understand the decisions, processes, and outcomes of the case study.

**R5** Replicability: Verner et al. [24] highlight "that any researchers wishing to replicate the study can do so in the knowledge that they are following the same protocol as that used in the original research" [24]. Runeson et al. highlight that " the replications as such should add to the validity of the research findings" [23]. Thus, a replicable case study will contribute to the quality of such.

**R6** Legal, Ethical, and Professional Requirements: Data collection, data processing, and analysis executed during the case study conduction must not defy legal, ethical, and professional requirements [23, p. 40]. According to Runeson et al., "an unethical and disreputable study can undermine the overall reputation of the discipline of software engineering research" [23].

### 5.3.2. Analysis-Specific Requirements

This section presents analysis-specific requirements **R10** to **R17**. These requirements ensure that the case study elaborates on all points of interest.

**R10** Uncertainty Categories: The case study validates that the UIA covers a wide range of categories of uncertainty, such as nature, level, and location [9, p. 3]. By considering a wide range the categories, the detailedness of the case study is improved.

**R11** Data Types: The case study validates that the UIA considers sensitive and non-sensitive data. Sensitive data may be personal, but also non-personal data [15]. This case study focuses on uncertainties that affect confidentiality. Confidentiality concerns only sensitive data, data that has to be protected from "unauthorized individuals, entities, or processes" [11]. This means that non-confidential data can

constrain propagation. Thus, the case study must evaluate if the UIA considers sensitive and non-sensitive data when analyzing the propagation.

**R12** Propagation Types: The case study considers structural and data flow propagation. Benkler's UIA propagates uncertainties structurally. However, he assumes that a UIA can narrow the *Impact Set* further by considering propagation alongside the data flow. Thus, the case study must evaluate whether the *Impact Set* can be narrowed by considering data flow propagation.

**R13** Usability of the UIA: The UIA must be usable. According to Benkler, architects must estimate the impact of each uncertainty and each element [4]. For complex architectures, the effort might be huge. The UIA aims to reduce the effort for architects, which is required to estimate the impact on confidentiality due to architectural uncertainties [4]. The smaller the number of components to examine, the lower the effort for the architects. Conversely, this means that if the UIA cannot reduce the number of such components, then the UIA is useless. Thus, the case study must investigate if the UIA is usable.

**R14** Functionality of the UIA: The UIA must (1) annotate components with uncertainties and (2) automatically propagate uncertainties. The rationale is trivial because an architect must be able to annotate components with uncertainties, and the UIA must provide results of the propagation analysis. Thus, the case study must verify that the program is functional.

**R15** Accuracy of the UIA: Results of the UIA must be accurate. Incorrect propagation results might degrade reliability. Thus, the case study must analyze the accuracy of the UIA.

**R16** Uncertainty Types in the Mobility Domain: The case study explores what uncertainty types can occur in the mobility domain and whether they differ from those derived from the healthcare domain by Benkler [4]. Benkler instantiated an uncertainty template for the healthcare domain during his case study. We are interested in the uncertainty template for the mobility domain.

**R17** Uncertainty Propagation in the Mobility Domain: The case study explores how uncertainties propagate in the mobility domain and whether the propagation differs from that in the healthcare domain. We want to explore uncertainties and their behavior in diverse domains regarding confidentiality.

## 5.4. Research Questions

In the previous section, we defined analysis-specific requirements. In this section, we derive from these requirements corresponding research questions. Considering research questions before conducting the case study helps maintain the focus on essential matters. It should be noted that the research questions might be refined during the case study conduction [23]. Runeson et al. define research questions as follows:

> "Research questions are statements about the knowledge that is being sought, or is expected to be discovered, during the case study. The discovery or attainment of this knowledge demonstrates that the case study has achieved its intended objectives." [23]

The research questions are divided into two groups. Section 5.4.1 illustrates the *exploratory group* that aims to explore uncertainties and their impact on confidentiality in the mobility domain. The *evaluatory group* aims to evaluate the UIA approach proposed by Benkler [4] and summarized in Section 5.4.2. The research question **ID** comprises a prefix **RQ** and a number for serial numbering. Runeson et al. further highlight, "Research questions may be organized into a hierarchy of more general and more specific research questions" [23]. We use this approach to structure the research questions. A nested research question contains a unique literal as a suffix.

### 5.4.1. Exploratory Research Questions

The exploratory research questions are derived from requirements **R16** and **R17**. Answers to these research questions help explore the mobility domain regarding uncertainties and their propagation regarding confidentiality.

**RQ1** Which uncertainties exist in the mobility domain? (**R16**)

**RQ2** How do uncertainties propagate and affect confidentiality in the domain? (**R17**)

### 5.4.2. Evaluatory Research Questions

The evaluatory research questions are derived from requirements **R13** to **R15**. The UIA is usable (**RQ3**) if it fulfills the promised simplifications. According to Benkler, the UIA should reduce manual effort and require expertise [4]. Furthermore, UIA aims to enable architects to annotate uncertainties in architectural elements (**RQ4a**). At the same time, the UIA automatically propagates the uncertainties and provides a set of possibly affected architectural elements, the so-called *Impact Set* (**RQ4b**). However, even if the previous questions provide positive answers, the accuracy of the propagation results plays a significant role in the overall usability (**RQ5**).

**RQ3** Is the UIA usable? (**R13**)

- **RQ3a** Does the UIA reduce the set of model elements that must be considered when analyzing the impact of uncertainties regarding confidentiality? [4, p. 87]

**RQ4** Is the UIA capable of supporting the uncertainty annotation and propagation? (**R14**)

- **RQ4a** Can architects annotate all architectural component types in the corresponding ADL?
- **RQ4b** Can the UIA determine the set of potentially affected architectural elements automatically? (Propagation)

**RQ5** Is the UIA accurate? (**R16**)

**RQ5a** Is the UIA precise and comprehensive?

**RQ5b** Is the affected set possibly larger than the impact set when considering the propagation alongside the data flow?

## 5.5. Classification of the Required Case Study

First of all, the rationales and objectives imply that we will "investigate [...] a contemporary software engineering phenomenon within its real-life context" [23]. Thus, the case study is the right research strategy. Further in this section, the units of analysis are defined, and the required case study is classified as suggested by Runeson et al. [23].

According to Verner et al., a "case study research may be classified based on its purpose into descriptive, explanatory, exploratory or evaluatory" [24]. The focus of the required case study is primarily to evaluate the completeness and correctness of the UIA approach. Primarily, we classify this case study as *evaluatory*. Following the second objective, this case study possesses an *exploratory* part. This part deals with the domain exploration regarding uncertainty types and their propagation in the mobility domain.

According to Yin, a case is "the main subject of study in a case study" [27]. However, he highlights two additional terms (1) the unit of analysis and (2) the embedded unit of analysis. Previously, we ascertained from the case study objectives that our case study is evaluatory and exploratory. This finding hints at having two units of analysis. The first unit of analysis is the mobility domain with the inherent uncertainty types and their propagation in the domain. The second unit of analysis is the UIA approach which helps researchers propagate uncertainties regarding confidentiality through the architecture. However, both units of analysis are investigated in the same real-life context, which is the uncertainty propagation through software architecture. Thus, our units of analysis are embedded. According to Runeson et al. and the case definition, this case study is an embedded single-case study [23].

The **RQ1** and **RQ2** belong to the first unit of analysis. The remaining research questions, **RQ3** to **RQ5**, belong to the second unit of analysis concerning the validity of the UIA.

## 5.6. Case Selection

"In case studies, the case and the units of analysis should be selected intentionally" [23]. Thus, in this section, we define the criteria for case selection. The Criterion ID comprises a prefix **C** for **C**riterion, followed by a unique number. Criteria for case selection:

**C1** Mobility Context: In our setting, the researcher must investigate the defined units of analysis in the mobility domain.

**C2** Availability of Extensive Information on the Domain: Architects require extensive information about the domain to derive *Architectural Design Decisions (ADDs)*, *Uncertainty Types* and construct an architectural model within a domain. Specifications, already existing models, well-documented software, and open-source software projects

might serve as information sources. Benkler used a well-documented open-source software project, the Corona Warn App (CWA), for quick model creation [4].

**C3** Embedded in a Real-World Context: The information sources need to be established and used in the real world. Thus, the investigators guarantee that the built case is a phenomenon within its real-life context. This criterion supports requirement **R1**.

## 5.7. Data Collection and Analysis

In this section, we propose processes to collect and analyze the required data to answer defined research questions. This section's first part defines the required data and propagation algorithms. The second part defines collecting and evaluating data for each research question.

### 5.7.1. Definitions

This section presents the definitions of the *Uncertainty*, *Affected*, and *Impact Set* (Section 5.7.1.1). Afterward, we present two approaches to derivate the *Impact Set* in Sections 5.7.1.2 and 5.7.1.3. The first approach derives the *Impact Set* following the structural uncertainty propagation. The second approach defines uncertainty propagation alongside the data flow. For the data analysis, we require a method of how to unite these two *Impact Sets*. Section 5.7.1.4 describes the merging strategy. Finally, we discuss the required data sources in Section 5.7.1.5.

#### 5.7.1.1. Uncertainty, Affected, and Impact Set

**Uncertainty Set** "A set of architectural elements on which uncertainties [...] have a direct impact" [4].

**Affected Set** An *Affected Set* contains architectural elements "which are actually affected" [4] by uncertainties. This set is independent of any UIA solution.

**Impact Set** A set of architectural elements on which uncertainties have a direct impact and on which they might have an indirect impact [4]. Note: The impact set depends on the impact analysis approach, e.g., structural propagation or data flow propagation. The *Impact Set* overestimates the *Affected Set* because the automated *Affected Set* derivation "can be reduced to the undecidable halting problem" [4].

**Relationship between the sets** The *Impact Set* is the most extensive. It contains the *Affected Set* because all affected architectural elements are also potentially affected. The *Uncertainty Set* is the smallest set because it only contains architectural elements that are directly affected. This set is part of the other sets. The following formula describes this relationship.

$$\textit{Uncertainty Set} \subseteq \textit{Affected Set} \subseteq \textit{Impact Set}$$

#### 5.7.1.2. Impact Set Derivation from Structural Propagation

The structural propagation is similar to the change impact propagation proposed for the Karlsruhe Architectural Maintainability Prediction (KAMP) project [22]. Benkler uses the KAMP approach to motivate his UIA solution [4, p. 64].

#### 5.7.1.3. Impact Set Derivation from Data Flow Propagation

Benkler considers only the structural propagation in his work [4]. However, he points out the propagation along identified data flows as a limitation of his work. Thus, we add the analysis of propagation along data flows.

In contrast, the impact set derivation from the data flow propagation does not have a solid foundation. Thus, for this case study, we define the impact set derivation according to the data flow. The author's little experience regarding uncertainty propagation and impact assessment on confidentiality threatens internal validity. Section 7.3 presents this threat in more detail.

The following list presents a simplified algorithm for deriving the *Impact Set* according to the data flow.

1. Add the annotated architectural element to the impact set. (Direct impact)

2. Identify possible data flows.

3. Follow each data flow that starts or traverses the directly impacted architectural element; add each architectural element that is traversed by the data flow to the impact set. (Indirect impact)

Experienced architects derive the *Affected Set* from this *Impact Set* similar to the approach proposed by Benkler for structural propagation [4].

#### 5.7.1.4. Merging Impact Sets

Suppose the structural impact set is $ImpactSet_{st}$ and $ImpactSet_{df}$ is the impact set derived from the data flow propagation.

$$ImpactSet = ImpactSet_{st} \cup ImpactSet_{df} \tag{5.1}$$

**Note:** Two architectural elements are equal if they have the same path and the same annotated uncertainty at the starting architectural element.

$$element_a = element_b \iff path_a = path_b \land uncertainty_a = uncertainty_b$$

Where $element_x \in ImpactSet$ is an architectural element affected by uncertainty. $path_x$ is the respective propagation path to this potentially affected architectural element $element_x$ and $uncertainty_x$ is the instantiated uncertainty affecting this $element_x$.

#### 5.7.1.5. Data Sources

Good data sources are essential to fulfill case study requirements. Verner et al. [24, p. 9] highlight six primary data sources. Documentation, physical artifacts, interviews, and others. Interviewing expert architects or domain experts can provide a solid foundation for the requirements. However, conducting extensive interviews with expert architects is not feasible in the scope of this thesis. We argue that documents and physical artifacts are sufficient as the primary source for the required case study. Good examples of documentation as a data source are documents from related work or reports from other case studies within a domain.

To explore the mobility domain and build an architectural model, the Mobility Data Specification (MDS) seems to be an adequate data source because of its broad and long-term use [13]. For the evaluatory part, we use the created architectural model in conjunction with documents about uncertainties, uncertainty propagation regarding confidentiality, and others. Hahner [9] and Benkler [4] are the primary consulted documents.

### 5.7.2. RQ1 – Uncertainty Types in the Mobility Domain

Recall that **RQ1** asks which uncertainties exist in the mobility domain. We must examine the mobility domain to answer this exploratory research question. We use the uncertainty-type derivation approach proposed by Benkler [4, p. 58]. This approach guides expert architects through the derivation process. Expert architects use their knowledge about the domain in conjunction with additional information regarding the domain to derive ADDs and, from these decisions, derive uncertainty types. Benkler validated his uncertainty-type derivation approach; however, independent researchers have not validated this approach to date. Due to time constraints for this thesis, it is impossible to validate this approach within the scope of this thesis. Thus, this approach threatens internal validity. Section 7.3 elaborates on this issue in more detail.

**Data Collection** As described previously, we must examine the mobility domain regarding ADDs. We decided not to conduct interviews with expert architects but rather consult literature concerning the mobility domain. According to Runeson et al., we require the *Third degree* of data collection methods. A systematic literature review has to be performed on the mobility domain. This collected data is qualitative.

We use the MDS proposed by the Open Mobility Foundation (OMF) [13] as the reference for the mobility domain and the already instantiated uncertainty template for the CWA case study [4]. Benkler assumes this template contains uncertainty types suitable for similar component-based architectures [4, p. 89]. This justifies the CWA uncertainty template as a starting point.

**Analysis** We apply the uncertainty type derivation process to derive ADDs from the collected data and, together with the *Reference Set*, instantiate an uncertainty template for the mobility domain. The instantiated uncertainty template is the artifact that answers **RQ1**.

However, we use the instantiated CWA uncertainty template in the initial conduction. Therefore, we derive the first ADDs and check if they are present in the instantiated

uncertainty template from the CWA case study. Each congruent finding supports Benkler's hypothesis that his proposed CWA uncertainty template could be suitable for other domains [4].

### 5.7.3. RQ2 – Uncertainty Propagation in the Mobility Domain

On a higher level, we are interested in whether domain-independent and domain-dependent uncertainties exist. However, we conduct a case study and cannot provide a general answer to such a question. We limit ourselves to the mobility domain and investigate how uncertainties propagate and affect confidentiality in the mobility domain (**RQ2**).

**Data Collection** To answer this question, we first create multiple scenarios that specify uncertainties and where they have a direct impact. These scenarios serve as input for the propagation path derivation. For the analysis, we require (1) the propagation paths determined from these scenarios by the UIA and (2) the manually determined propagation paths to potentially affected elements from these scenarios. We only use structural propagation for manual propagation because we aim to investigate how these two propagation approaches differ. This data will help to answer how the uncertainties propagate and affect confidentiality in this domain. According to Runeson et al. [23], the required data collection method is direct, i.e., of *First degree*. The collected data is quantitative because the *Impact* and *Affected Sets* contain classified architectural elements.

**Analysis** Benkler's UIA was designed to be domain-independent. As mentioned earlier, Benkler validated the UIA within the cross-domain healthcare and digital contact tracing and achieved good results, see Section 3.1. We compare the two determined propagation paths and discuss them. If the paths are equal, then it is a sign that the uncertainty propagation behaves similarly in the cross-domain and the mobility domain. This fact could indicate that the respective uncertainty type is domain-independent. The reverse indicates that the respective uncertainty type is domain-dependent.

### 5.7.4. RQ3 – UIA Usability

Recall that **RQ3** concerns the usability of the UIA. According to requirement **R13**, this is the case if the UIA reduces the number of architectural elements to review if the elements are actually affected (**RQ3a**).

**Data Collection (RQ3a)** For **RQ3a**, we will use Benkler's approach to evaluate the degree of the set reduction. We require the total number of architectural elements an architect must review for a scenario. Usually, this is the total number of all architectural elements of the analyzed model because each element must be reviewed. The second data is the *Impact Set* given by the UIA and manual derivation for the same scenario. This data is directly collected (*First degree*) and is quantitative.

**Evaluation (RQ3a)** For **RQ3a**, we use the evaluation procedure proposed by Benkler [4, p. 97]. $r$ is the ratio where $ImpactSet$ is the number of elements in the *Impact Set*, and $n$ is the number of all elements present in a model. Calculating these values for the manual

evaluation and the automated approach (by UIA) allows us to compare the required effort for assessing the impact. Ration $r < 1$ indicates a reduction of analysis effort.

$$r_{\text{manual}} = \frac{ImpactSet_{manual}}{n} \tag{5.2}$$

$$r_{\text{uia}} = \frac{ImpactSet_{uia}}{n} \tag{5.3}$$

Of course, this research question must be evaluated while considering the accuracy because the *Impact Set* might be reduced; however, not comprehensive. Research question **RQ5** deals with UIA accuracy.

### 5.7.5. RQ4 – UIA Functionality

Recall research question **RQ4** breaks down into research questions **RQ4a** and **RQ4b**. **RQ4a** asks if the UIA can enable architects to annotate all architectural concept types according to Component-Based Software Engineering (CBSE). **RQ4b** asks if the UIA can propagate uncertainties and determine the impact set.

**Data Collection (RQ4a)** To evaluate **RQ4a**, we require the total number of available Palladio concept types within the architectural model (*total number of Palladio concept types*). We limit ourselves to Palladio concept types because our model is based on Palladio. Further, we must derivate how many of these elements are annotatable. For that, we create a scenario that contains an instantiated uncertainty per the Palladio concept type. For the scenario, used uncertainty types must at least specify the *Assignable Element Type*. Furthermore, the uncertainty template that contains these uncertainty types must provide at least as many types as there are different Palladio concept types in the model. We count the number of uncertainty types that can be instantiated without failing (*annotatable Palladio component types*). We estimate the method of *First degree* because of the high control over the collected data while the data is quantitative.

**Evaluation (RQ4a)** The data collection provides the *total number of Palladio concept types* and *annotatable Palladio component types* as input for Equation (5.4). *Annotation Completeness* = 1 is desired and implies that the UIA can annotate uncertainties of all types.

$$\text{Annotation Completeness} = \frac{\text{annotatable Palladio concept types}}{\text{total number of Palladio concept types}} \tag{5.4}$$

**Data Collection (RQ4b)** Also, during data collection, we limit ourselves to Palladio concept types because our model is based on Palladio. For research question **RQ4b**, we first require the *total number of Palladio concept types*. We collect further required data from the UIA codebase. We determine code passages dealing with uncertainty propagation algorithms and count the number of *implemented propagation algorithms*. We estimate the method to be of *First degree* because of the high control over the collected data while the data is quantitative.

**Evaluation (RQ4b)** For a pair of Palladio concept types $A$ and $B$, the structural propagations require two algorithms: $A \rightarrow B$ and $B \rightarrow A$ [4]. According to Benkler, $A \rightarrow A$ does not propagate. He assumes this because the UIA executes only one iteration. However, we want to allow such propagations and calculate the *number of required propagation algorithms* as follows:

$$\text{number of required propagation algorithms} = \text{total number of Palladio concept types}^2$$

Now we have all the required data and use Equation (5.5) to examine *Propagation Algorithm Coverage*. *Propagation Algorithm Coverage* = 1 is desired and implies that the UIA can structurally propagate uncertainties of all types.

$$\text{Propagation Algorithm Coverage} = \frac{\text{implemented propagation algorithms}}{\text{number of required propagation algorithms}} \quad (5.5)$$

### 5.7.6. RQ5 – UIA Accuracy

The UIA derives an overestimated set of architectural elements: the *Impact Set*. This set must be as accurate as possible to reduce the estimation effort. On the other side, it must not be too small because architects must be able to rely on the set for further uncertainty estimation without missing out on any potentially affected elements. Thus, **RQ5** asks if the UIA is accurate. This research question is subdivided into two subquestions. **RQ5a** asks if the UIA is precise and comprehensive, while **RQ5b** asks if the *Impact Set* might miss affected architectural elements.

**Data Collection (RQ5a)** The *Impact Set* must be close to the *Affected Set* because the UIA should reduce the analysis work but still contain at least all affected elements. According to the *Affected Set* definition, the *Impact Set* must be close to the *Affected Set*. Thus, we use the *Affected Set* as the gold standard.

We require the *Impact Set* and the *Affected Set*. For that, we create multiple scenarios and analyze these automatically and manually. With the UIA we determine the $ImpactSet_{UIA}$. Then we manually derive the $ImpactSet_{manual}$, as defined in Section 5.7.1. The manual derivation considers both structural propagation and data flow propagation. We apply mobility domain-specific literature and the model definition (Section 6.3.2) to the $ImpactSet_{manual}$ to derive the *Affected Set*. Qualitative (domain knowledge) and quantitative (determined sets) are required data categories.

Please note that the manual derivation of the *Impact Set* and the *Affected Set* is difficult and requires knowledge and experience. The author possesses little experience; this might threaten the internal validity. This fact is outlined in more detail and how we deal with it in Section 7.3.

**Evaluation (RQ5a)** Benkler evaluates accuracy using the *Precision and Recall* method. We keep using this approach. Definitions and equations below declare how to calculate $Precision$, $Recall$, and $Accuracy$.

For the *Precision*, *Recall*, and *Accuracy* metrics, we define the following variables:

- *FP*:= Type I Error: false positive
  $x \in FP : x \in ImpactSet_{UIA} \wedge x \notin AffectedSet$
- *FN*:= Type II Error: false negative
  $x \in FN : x \notin ImpactSet_{UIA} \wedge x \in AffectedSet$
- *TP*:= Valid: true positive
  $x \in TP : x \in ImpactSet_{UIA} \wedge x \in AffectedSet$
- *TN*:= Valid: true negative
  $x \in TN : x \notin ImpactSet_{UIA} \wedge x \notin AffectedSet$

**Precision**

$$Precision = \frac{TP}{TP + FP}$$

**Recall**

$$Recall = \frac{TP}{TP + FN}$$

**Accuracy**

$$Accuracy = \frac{TP + TN}{Total}$$

**Data Collection (RQ5b)** The answer to research question **RQ5b** builds upon the analysis of **RQ5a**. We require the calculated *Recall* from the previous sub-research question.

**Evaluation (RQ5b)** From the calculated *Recall*, we can conclude if the automatically determined *Impact Set* misses affected elements. This is the case if the $Recall < 1$ because the UIA derived false negatives.

## 5.8. Legal, Ethical, and Professional Considerations

Runeson et al. state that researchers should consider legal, ethical, and professional requirements during the case study design to prevent illegal actions and unethical, unprofessional behavior [23].

Our case study does not require any individual participants. Therefore, most of the threatening factors do not occur in this case study. For example, we do not conduct interviews and collect data that could divulge the identity of participants. The sensitive data we speak about during the case study refers to a model of such data but not concrete data of individuals. Furthermore, this case study does not produce any sensitive results that could prevent reporting.

The main legal document we must consider is the General Data Protection Regulation (GDPR). According to Art. 2 GDPR, this regulation does not apply because no personal data is processed as described in the previous paragraph. We have no further qualms about making the data available to the public.

The research does not require any confidential data from any organization; thus, no non-disclosure agreements are required. As previously mentioned, no individuals participated in the case study, so no informed consent is required.

# 6. Mobility Case Study

This chapter presents a mobility case study for validating the Uncertainty Impact Analysis (UIA) regarding confidentiality. This case study is aligned with the case study design proposed in Chapter 5 and executed according to the proposed investigation process in Chapter 4.

First, we clarify the roles of the investigators in Section 6.1. Section 6.2 elaborates on the case, while the consecutive section, Section 6.3, describes the instantiated model and justifies it. The essential part of the case study, data collection and analysis, is conducted in Section 6.4 and 6.5. In order to reduce redundancies, the limitations of this case study are discussed in Chapter 7 in Section 7.4.

## 6.1. Roles

Chapter 4 presented four roles: *Investigator*, *Analyst*, *Expert Architect*, and *User Architect*. Each role executes a particular step of the investigation process. According to our role definition, all participants in this case study gain the *Investigator* role. Due to the organizational setting of this case study, which is mainly characterized by the limited resources of a Bachelor's thesis, we allocate the roles as follows:

**Denis Priss** is the leading *Investigator*; he is also an *Analyst*, *Expert Architect*, and *User Architect*. He is responsible for the organizational part, requirements definition, and case study design. Furthermore, he elaborates on the data collection and analysis and reports to Sebastian Hahner and Maximilian Walter each week in the scope of his thesis.

**Sebastian Hahner** is the second *Investigator*. He is the supervisor responsible for the case study's organizational part. He reviews the documentation and the created artifacts. Further, Sebastian Hahner is an *Expert Architect* regarding Component-Based Software Engineering (CBSE) and uncertainty propagation; however, he only acts in an advisory capacity.

**Maximilian Walter** is the third *Investigator*. He is the supervisor responsible for the case study's organizational part. He reviews the documentation and the created artifacts. Further, Maximilian Walter is an *Expert Architect* regarding CBSE; however, he only acts in an advisory capacity.

## 6.2. Case Selection

In Section 5.6, we defined the criteria for case selection. To fulfill these requirements, we base our case on the Mobility Data Specification (MDS) [13] proposed by the Open

Mobility Foundation (OMF). The MDS represents the mobility domain (**C1**) and provides extensive information about the domain (**C2**). Open Mobility Foundation (OMF) provides general information about the MDS [13], schemas for endpoint specifications [14], privacy guide for cities [15], MDS under General Data Protection Regulation (GDPR) [17], and furthermore. MDS is well-known and established in the real world; this ensures proximity to reality (**C3**).

We use the *Use Case Database* [16] maintained by the Open Mobility Foundation (OMF) to derive our first case. Two interesting examples are *Restricted Area Rides* and *Top Speed Calculations*. Louisville implements both examples (**C3**). With the first example, the city searches for "locations where devices are operating or passing through restricted areas." [16]. With the second example, they "determine the average speed of a trip and ensure it meets requirements of top speed and slow area requirements." [16]. We first focus on these two examples because they are intuitive, used in the real world, and require less implementation effort.

According to these examples, we require two parties: a city and a mobility provider. The mobility provider maintains vehicles and provides mobility services to the public. The vehicles push vehicle-specific data, such as location and speed, to the mobility provider. The mobility provider processes and stores user data and grants registered users vehicle access. According to those two examples, the city demands to provide citizens with the findings. However, the city does not possess the required data to generate statistics. For that, the mobility provider implements the *Provider*-Application Programming Interface (API) specified by the MDS [14].

The architectural model this introduced case. During the design phase, they gather knowledge about the mobility domain, make Architectural Design Decisions (ADDs), and consider uncertainties, their propagation, and their impact on confidentiality.

## 6.3. Modeling

This section describes the designed architectural model, used technologies, and tools. We start with the technologies and tools in Section 6.3.1 and finalize this section with the description of the architectural model in Section 6.3.2.

### 6.3.1. Technologies and Tools

The UIA proposed by Benkler and presented in Section 2.4.2 is a Palladio add-on [4]. As described in Section 2.3, Palladio comprises several parts; one of those is the Palladio Component Model (PCM), a meta-model for modeling component-based architectures. This meta-model is based on the Eclipse Modeling Framework (EMF). Thus, we use UIA, Palladio, and the Eclipse IDE (Eclipse Modeling Project). UIA requires additional dependencies, which we do not list here because they are less relevant for the modeling phase.

### 6.3.2. Model

The MDS is designed to improve communication between cities and mobility providers. We presented this principle in Section 2.2. Therefore, we start with two basic components *Agency* and *Provider*. The *City* and *Company*, respectively, implement these components. *City* and *Company* are resource containers that require a database to store processed data, *AgencyDB* and *ProviderDB*.

Please note: MDS uses *Agency* and *Provider* as API identifiers. However, we refer to MDS APIs by prefixing an *I* before the identifier. The *I* stands for an interface, e.i., *IAgency* or *IProvider*.

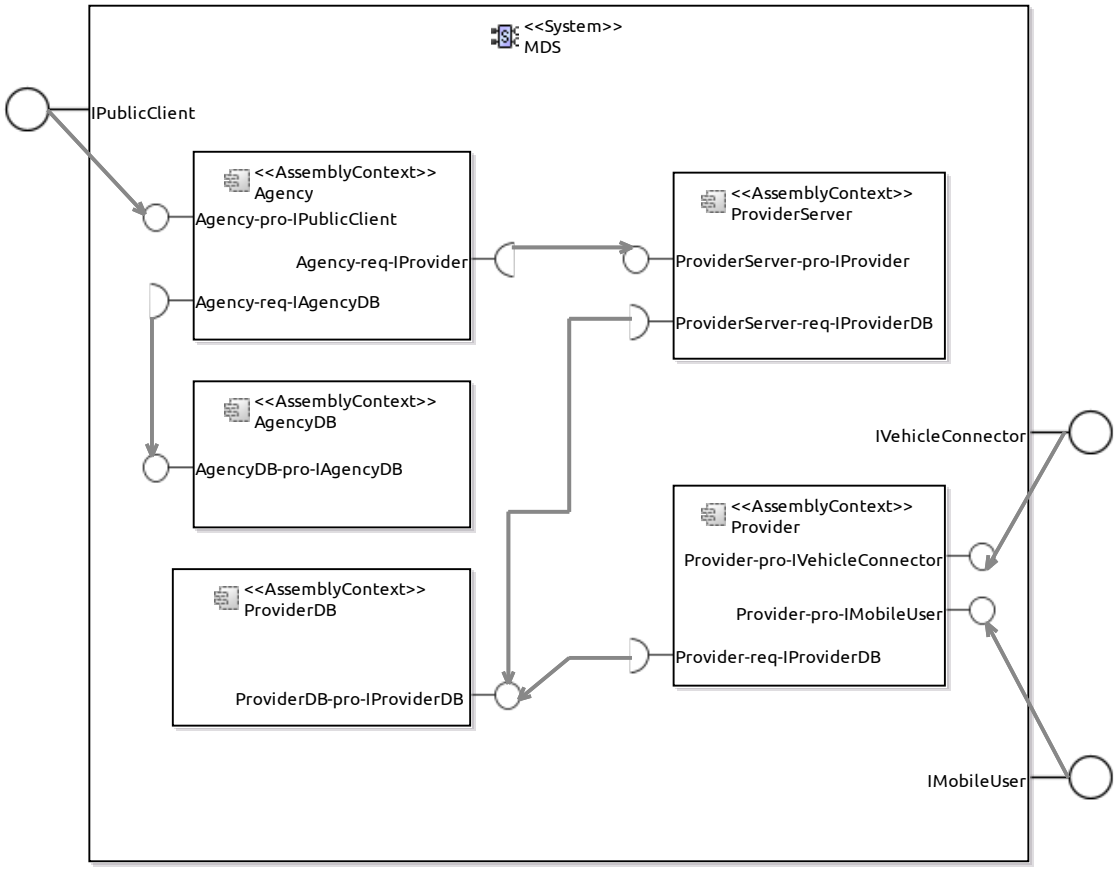


Figure 6.1.: Assembly diagram for the MDS-System.

During the case selection, we identified two interesting examples: (1) *Restricted Area Rides* and (2) *Top Speed Calculations*. Both examples require the *Provider* API defined by the MDS [14]. To enable communication between the city and the mobility provider, we introduce the *ProviderClient* component, which exposes the required interface *IProvider*. The *Agency* queries the data through this interface, processes the data, and saves the results into the *AgencyDB*. The *ProviderClient* is responsible for data provision. Therefore, it queries the data from the *ProviderDB*.

The *Provider* component performs data processing for user services, e.g., user registration and vehicle rental. Therefore, the *Provider* exposes the *IMobileUser* interface and saves the processed data into the *ProviderDB*. Furthermore, the *Provider* exposes the *IVehicleConnector*, which is used by the vehicles to transfer vehicle-specific data. This data is also processed by the *Provider* and saved into the *ProviderDB*. Finally, the *Agency* exposes the *IPublicClient* interface, which interested individuals might use to view the statistics about average speed and "locations where devices are operating or passing through restricted areas." [16].

Figure 6.1 illustrates the assembly diagram of the described MDS System. Further model diagrams are available in Appendix A. The presented model is created during the first iteration.

## 6.4. Data Collection

In this section, we collect evidence, as proposed in Section 5.7. Section 6.4.1 describes found ADDs and which uncertainty types exist in the mobility domain. Section 6.4.2 examines the model introduced in Section 6.3.2 regarding the number of used Palladio concepts and Palladio concept types. Implemented propagation algorithms are examined in Section 6.4.3. Section 6.4.4 and Section 6.4.5 present two scenarios and determined data from these scenarios. Scenario S.03 examines the annotation completeness and is presented in Section 6.4.6.

### 6.4.1. Uncertainty Types in the Mobility Domain

According to MDS, the mobility provider possesses vehicle-specific data, e.g., location and speed, and user-specific data required to provide mobility services to citizens. This data is stored in a database. However, the MDS does not specify which data the Provider gathers. MDS specifies only data that the mobility Provider can share with the city. Thus, the ADD must be made about the data stored in the Provider's database. *What data is persisted?* is an uncertainty type in the mobility domain. The characteristics of this uncertainty type exist in the Corona Warn App (CWA) uncertainty template created by Benkler [5].

As already mentioned, MDS specifies APIs for communication between cities and mobility providers. However, as discussed in the previous paragraph, MDS does not specify APIs besides those mentioned, e.g., there is no specification on an API for user registration. Thus, the uncertainty type *What is the structure of the interface?* exists not only in the healthcare domain but also in the mobility domain. The CWA uncertainty template lists this type [5].

Due to the authors' experience, every developer creates bugs while producing code. Thus, a component might be insecure due to software bugs. Benkler lists this uncertainty type in the CWA uncertainty template.

### 6.4.2. Architectural Elements

For further evaluation, we must determine the total number of architectural elements and architectural element types. We mentioned previously that our model is based on PCM. Thus, we count the Palladio concept types and Palladio concepts. The model we refer to is described in Section 6.3.2. In this model, we count the following component types and the number of their instances.

- Resource Containers: 2
- Communication Links: 1
- Systems: 1
- Allocations: 5
- Assembly Contexts: 5
- Assembly Connectors: 7
- Operation Provided Roles: 9 (3 × system-level roles)
- Operation Required Roles: 4
- Operation Interfaces: 6
- Basic Components: 5
- Basic Components Behavior: 10
- Entry Level System Call: 1

In total: The model contains **56** Palladio concepts and **12** Palladio concept types.

### 6.4.3. Implemented Propagation Algorithms

Table 6.1 illustrates a matrix of the implemented propagation algorithms. As described in Section 5.7.5, we determined the data for the table from UIA's codebase [6]. The architectural elements in the first column indicate the starting point of the propagation (direct impacted architectural elements). The architectural elements in the first row indicate the possibly impacted architectural elements. Cells marked by an $x$ in the matrix indicate an implemented propagation algorithm. We created the following matrix from the UIA code base.

The number of $x$ in Table 6.1 represents the number of implemented propagation algorithms.

### 6.4.4. Scenario S.01 – What Data is Persisted on ProviderDB

In Section 6.4.1 the expert architect derived the uncertainty type *What data is persisted?*. Also, the expert architect described that this uncertainty type might occur on the *ProviderDB* component. In this scenario we annotate the *ProviderDB* basic component with this uncertainty.

Table 6.1.: Matrix of implemented propagation algorithms.

| Architectural Element Type | System | Hardware Resource | Basic Component Type | Component Instance | Basic Component Behavior | Communication Component | Communication Resource | System Interface | Component Interface Instance | Component Interface Type | Usage Behavior |
|---|---|---|---|---|---|---|---|---|---|---|---|
| System | | | | | | x | | x | x | x | |
| Hardware Resource | | | | x | | x | | | | | |
| Basic Component Type | | x | | | | | | x | | | |
| Component Instance | | | | | | | | | | | |
| Basic Component Behavior | | | | | | | | x | | | |
| Communication Component | | x | | | | | | | | | |
| Communication Resource | | | | | | | | | | | |
| System Interface | | | | | | | | | | x | |
| Component Interface Instance | | | | | | | | | | | |
| Component Interface Type | | | | | | | | x | | | |
| Usage Behavior | | | | | | | | | x | | |

#### Tested Uncertainty Types and their Annotation

- CU2: *What Data is Persisted?*
  - Assignable to: *Basic Component Type*
  - Annotated to: *ProviderDB* `(_trKiIBf1Ee22hrP83ESheA)`
  - Has structural impact on: *Hardware Resource*

**Automated Uncertainty Impact Analysis** UIA produces the propagation path shown in Listing 6.1.

Listing 6.1: Automated uncertainty propagation results for *What data is persisted?*

```
ID: _-pqAsCelEe21k8KBakoDiw
Name: What data is persisted on Provider DB
Propagated to:
ProviderDB(_trKiIBf1Ee22hrP83ESheA): BasicComponent
         ---
Company(_PY9tYA3NEe2B174O52ER9A): ResourceContainer
        ProviderDB(_trKiIBf1Ee22hrP83ESheA): BasicComponent
        ProviderDB(_BqVw4Bf3Ee22hrP83ESheA): AssemblyContext
        ProviderDB(_toDcYBf4Ee22hrP83ESheA): AllocationContext
        Company(_PY9tYA3NEe2B174O52ER9A): ResourceContainer
```

Impact Set from automated propagation:

- ProviderDB `(_trKiIBf1Ee22hrP83ESheA)` : BasicComponent
- Company `(_PY9tYA3NEe2B174O52ER9A)` : ResourceContainer

**Manual Uncertainty Impact Analysis** Like the impact set, the affected set trivially includes the annotated architectural element [4, 65 ff.]. In our case, it is the *ProviderDB* basic component. Further, we determine the architectural elements that might suffer from indirect impact. We identify the resource Container *Company* as part of the affected set because if the database stores sensitive data, the location of the hardware is crucial to confidentiality. Furthermore, the interface of the *ProviderDB* (*Component Interface Type*) may be affected by this uncertainty. If *ProviderDB* stores confidential data, the interface requires proper security, and the communication must be encrypted. The *ProviderDB* is connected to the *ProviderClient* and the *Provider* assembly context, meaning they both might be affected. However, in our model, all these components are allocated to the same hardware resource; therefore, they are unaffected.

Following the data flow, we see that the data from *ProviderDB* flows across the entire system, and so does the impact of the uncertainty. Thus, the manually derived impact set contains all architectural elements present in our model.

$$|ImpactSet_{manual}| = 56$$

Still, because in this scenario the instantiated interfaces are known, the resulting affected set contains only these two following components:

- ProviderDB `(_trKiIBf1Ee22hrP83ESheA)` : BasicComponent
- Company `(_PY9tYA3NEe2B174O52ER9A)` : ResourceContainer

### 6.4.5. Scenario S.02 – Bugs and Uncertainty at Interface

In this scenario, we focus on the Agency component. We assume this component is insecure due to software bugs, and its interface is not yet fully worked out.

**Tested uncertainty types:**

- CU8: *What is the structure of the interface?*
  - Assignable to: *Component Interface Type*
  - Annotated to: *Agency-pro-IAgency* `(_osK0ABeyEe2_NIc9ghIYJQ)`
  - Has structural impact on: *none*
- CU27: *Is the component insecure due to software bugs?*
  - Assignable to *Basic Component Type*
  - Annotated to: *Agency* `(_cHvDEBeyEe2_NIc9ghIYJQ)`
  - Has structural impact on: *none*

**Automated Uncertainty Impact Analysis** UIA produces the propagation path shown in Listing 6.2.

Impact Set from automated propagation:

- Agency `(_cHvDEBeyEe2_NIc9ghIYJQ)` : BasicComponent
- Agency-pro-IAgency `(_osK0ABeyEe2_NIc9ghIYJQ)` : OperationProvidedRole

Listing 6.2: Automated uncertainty propagation results for *Scenario S.02*

```
ID: _ayY3oCq0Ee2LJKlApN9icw
Name: Agency component insecure due to software bugs
Propagated to:
Agency(_cHvDEBeyEe2_NIc9ghIYJQ): BasicComponent
         ---


NEXT UNCERTAINTY
ID: _qOvbIE5GEe2fDbqgNFoZJA
Name: Uncertain structure of IProvider interface
Propagated to:
Agency-req-IProvider(_Lr4MME46Ee2fDbqgNFoZJA): OperationRequiredRole
         ---
```

**Manual Uncertainty Impact Analysis** Like the impact set, the affected set trivially includes the directly annotated architectural elements [4, 65 ff.]. In our case, the *Agency* basic component and the *Agency-pro-IAgency* operation provided role are directly affected.

Further, we determine the architectural elements that might suffer from indirect impact. Suppose a component is insecure due to software bugs. An unauthorized individual might exploit this bug if the component exposes an entry point to access the hardware resource. In our case, *Agency* is accessible through *IPublicClient*. Thus, all components allocated to this resource are potentially affected. The *Agency* and the *AgencyDB* components are allocated to the *City* resource. However, the structure of the interface *Agency-pro-IAgency* is uncertain. This further might affect the *ProviderClient-req-IAgency* interface and the *ProviderClient* component.

These structurally derived potentially affected elements are still potentially affected even if we follow the data flow. Additionally, we might have data flow between all components allocated to the *Company* resource container. Thus, the entire system might be affected, and the impact set contains all elements from the model.

$$|ImpactSet_{manual}| = 56$$

On the other side, we know the structure of *ProviderServer-pro-IProvider*. We also know that data flows from *ProviderSever* to *Agency*. Thus, the uncertainty does not propagate further to the *Company* resource container. The affected set contains only elements that exist on the *Agency* resource container.

Affected Set contains eight architectural elements:

- Agency `(_cHvDEBeyEe2_NIc9ghIYJQ)` : BasicComponent
- Agency-req-IProvider `(_Lr4MME46Ee2fDbqgNFoZJA)` : OperationProvidedRole
- Agency-pro-IPublicClient `(_JfVQoBfKEe2_NIc9ghIYJQ)` : OperationProvidedRole
- Agency-req-IAgencyDB `(_tgWToE43Ee2fDbqgNFoZJA)` : OperationProvidedRole
- AgencyDB-pro-IAgencyDB `(_o61hEE43Ee2fDbqgNFoZJA)` : OperationProvidedRole
- IAgencyDB `(_hInh8E43Ee2fDbqgNFoZJA)` : OperationInterface
- AgencyDB `(_oBHNgE5oEe2fDbqgNFoZJA)` : AssemblyContext
- AgencyDB `(_j_K4UE43Ee2fDbqgNFoZJA)` : BasicComponent

- City `(_RszYQA3NEe2B174O52ER9A)` : ResourceContainer
- IPublicClient `(_fPveYBfJEe2_NIc9ghIYJQ)` : OperationInterface
- IPublicClient `(_YcZ7YB1FEe2VV5jU6VdTTg)` : OperationProvidedRole

### 6.4.6. Scenario S.03 – Annotation Completeness

In this scenario, we want to check how many Palladio concept types are annotatable. Benkler instantiated a `sample.uncertaintytemplate` that contains test uncertainty types for each Palladio concept type. These uncertainty types are not fully specified. They specify the *Assignable Element Type*, which is perfectly suitable for this scenario.

As presented in Section 6.4.2, our model contains 12 Palladio concept types. We use Benkler's sample uncertainty template and annotate each Palladio concept type once.

We could annotate all available concept types. The created uncertainty model is saved as `annotationCompleteness.uncertainty` .

## 6.5. Analysis

In this section, we use the data collected in Section 6.4 and analyze this data to answer research questions **RQ1** to **RQ5**. The analysis is executed according to the design introduced in Section 5.7. The following sections analyze the data and discuss the findings sequentially to the reaches questions.

### 6.5.1. RQ1 – Uncertainty Types in the Mobility Domain

In Section 6.4.1 , we found three uncertainty types for the mobility domain. As we pointed out in that section, all three types exist in the healthcare domain. Further iterations are required to find more uncertainty types and check if Benkler's assumption can sustain that the filled uncertainty template might be "suitable for similar CBSA" [4].

### 6.5.2. RQ2 – Uncertainty Propagation in the Mobility Domain

In Section 6.4.4, the scenario **S.01** produced results that are equal for the manually derived impact set and the set derived by the UIA. On the other side, scenario **S.02** revealed different results. According to the manual derivation, the uncertainty regarding bugs at the *Agency* component impacted on many other components. This might hint that this uncertainty type propagates differently in the mobility domain. However, we assume that there is an implementation issue with the UIA.

### 6.5.3. RQ3 – UIA Usability

**RQ3** concerns the usability of the UIA. According to requirement **R13**, this is the case if the UIA reduces the number of architectural elements to review (**RQ3a**). To analyze if the UIA is usable, we use the collected data from scenarios **S.01** and **S.02** and apply the evaluation strategy defined in Section 5.7.4.

Table 6.2.: Impact sets and rations for Scenarios S.01 and S.02.

| | S.01 | | S.02 | |
|---|---|---|---|---|
| | manual | UIA | manual | UIA |
| *ImpactSet* | 56 | 2 | 56 | 2 |
| *n* | 56 | 56 | 56 | 56 |
| *r* | 1.0 | 0.0357 | 1.0 | 0.0357 |

According to the ratio shown in Table 6.2, in both scenarios, the architects must assess only 4% of the software architecture if they use the UIA. This is a reduction of the required effort. However, this result is unreliable because linking results from Section 6.5.5 indicates that the affected set might be greater than the impact set derived by the UIA. Thus, architects could miss affected elements because the UIA did not list them as potentially affected.

### 6.5.4. RQ4 – UIA Functionality

Recall research question **RQ4** breaks down into research questions **RQ4a** and **RQ4b**. **RQ4a** asks if the UIA can enable architects to annotate all architectural concept types. **RQ4b** asks if the UIA can propagate uncertainties and determine the impact set.

#### RQ4a – Annotation Completeness

For the analysis of **RQ4a**, we use the collected data from scenario **S.03** and apply the evaluation strategy defined in Section 5.7.5.

As stated in Section 6.4.6, all found element types (12) in our model could be annotated with uncertainty types. Thus, we reach 100% coverage, meaning the UIA is complete regarding uncertainty annotation.

$$\text{Annotation Completeness} = \frac{12}{12} = 1 = 100\%$$

#### RQ4b – Implemented Propagation Algorithms

For the analysis of **RQ4b**, we use the collected data in Section 6.4.3 and apply the evaluation strategy defined in Section 5.7.5.

According to our data, the UIA implements 13 propagation algorithms. Table 6.1 shows the implemented algorithms. We found 12 architectural element types during our data collection. This means $12 * 12 = 144$ required propagation algorithms.

$$\text{Propagation Algorithm Coverage} = \frac{13}{144} = 0.0903 = 9.03\%$$

This analysis shows that only 9% of the required propagation algorithms are implemented. However, this result is still too optimistic because we did not consider the required

propagation algorithms along the data flow. However, the accuracy results from the next section imply that structural propagation of uncertainties alone is insufficient to provide software architects with a reliable starting point for further confidentiality analyses.

### 6.5.5. RQ5 – UIA Accuracy

**RQ5** asks if the UIA is accurate. This research question is subdivided into two subquestions. **RQ5a** asks if the UIA is precise and comprehensive, while **RQ5b** asks if the *Impact Set* might miss affected architectural elements.

To analyze if the UIA is accurate, we use the collected data from scenarios **S.01** and **S.02** and apply the evaluation strategy defined in Section 5.7.6. Table 6.3 presents a results overview. The abbreviations have the following meaning:

**FP** Type I Error: false positive
**FN** Type II Error: false negative
**TP** Valid: true positive
**TN** Valid: true negative

Table 6.3.: Precision, recall, and accuracy for Scenarios S.01 and S.02.

| Scenario | Precision | Recall | Accuracy | TP | TN | FP | FN |
|---|---|---|---|---|---|---|---|
| S.01 | 1.0000 | 1.0000 | 1.0000 | 2 | 54 | 0 | 0 |
| S.02 | 0.5000 | 0.0909 | 0.8036 | 1 | 44 | 1 | 10 |

Scenario **S.01** has perfect results. However, we instantiated only a single uncertainty type on one architectural element.

For scenario **S0.2**, the precision reaches 50%, meaning the UIA recognized unaffected architectural elements as possibly affected. As discussed earlier, this is not a bad sign because the impact set should be overestimated. However, a recall of 9% highlights that in scenario **S.02** , the UIA recognized only 9% of the actually affected architectural elements. This contradicts the statement that the automatically derived impact set is an overestimation. It also provides an answer to **RQ5b**. The UIA estimates 80.36% of all present architectural elements as correct.

In conclusion, scenario **S.02** suggests that the UIA is not sufficiently accurate. Further iterations with more complex scenarios must be executed to evaluate this suggestion. Furthermore, according to precision, we assume that structural propagation of uncertainties alone is insufficient to provide software architects with a reliable starting point for further confidentiality analyses.

# 7. Evaluation

In this chapter, we evaluate our approach. We assess the quality of the case study, the investigation process, and the findings coverage of the case study. For this we use the Goal-Question-Metric (GQM) approach [2]. The GQM approach defines the measurements in a top-down fashion. Basili, Caldiera, and Rombach argue that for a purposeful evaluation, first, the researcher must define goals; second, derivate questions from the goals; and last, propose metrics to answer the questions. The top-down approach helps the researcher to focus on the essence and to avoid collecting and evaluating not required aspects, as well as not missing out on essential aspects.

Section 7.1 represents the evaluation plan and design. This section defines goals, questions, and metrics according to the GQM-Plan [1]. By applying this plan, we evaluate our approach and discuss the results in Section 7.2. Section 7.3 presents threats to validity and how we handle them. In Section 7.4, we discuss the assumptions and limitations of our work. The final section of this chapter, Section 7.5, links to the required information that enables researchers to reproduce the evaluation.

## 7.1. Evaluation Design

This section provides a set of goals, questions, and metrics according to the GQM-Plan [1]. The GQM approach proposed by Basili, Caldiera, and Rombach [1] enables us to evaluate the quality of the conducted mobility case study and improve it through iterations. GQM provides a top-down approach for the derivation of measurement techniques, namely (1) Goals, (2) Questions, and (3) Metrics [1]. After goal identification, questions are derived "to characterize the object of measurement" [1]. These characterizations provide a viewpoint from the quantitative perspective and allow metrics to be defined for purposeful measurement.

Table 7.1 presents an overview of the derived goals, the requirements defined in Section 5.3, and a mapping between those two. Requirements **R2** to **R5** induce **G1**. Here, the goal is to have a feasible and profound process that guides investigators through the case study conduction. **G2** incorporates defined requirements **R1** to **R6**. The goal is to produce a qualitative case study that fulfills the requirements of a "good case study" [23]. Requirements **R10** - **R17** are case study analysis-specific requirements. They reflect on the case study design and the case study analysis. Therefore, we join these requirements in goal **G3** and use them to evaluate the quality of the analysis step in this chapter.

The sections below represent the defined goals. We establish the following homogeneous structure for each goal: First, we describe the goal in more detail. Additionally, we present a table with the goal specification according to the approach proposed by Basili, Caldiera, and Rombach [1]. This table contains the *Purpose*, *Issue*, *Object*, and *Viewpoint*. The

Table 7.1.: Overview of goals and mapping to requirements.

| Goal ID | Goal | Reflected Requirements |
|---|---|---|
| G1 | **Qualitative investigation process**<br>The investigation process shall be profound, feasible, and provide expressive case study results. | R2 - R5 |
| G2 | **Qualitative case study**<br>The produced case study shall fulfill the requirements of a "good case study" [23]. | R1 - R6 |
| G3 | **Case study comprehensiveness**<br>The case study shall investigate all crucial points of interest. | R10 - R17 |

remaining content of the goal section consists of questions and metrics. Each question contains a short title, explanation, metrics, and guidance on how to proceed to answer the question. Finally, a note if required.

### 7.1.1. G1 – Qualitative Investigation Process

As described in Section 7.1, we aim to provide a process for investigators to guide them through the case study conduction and ensure a qualitative case study. Therefore, the process must be profound, feasible, and produce meaningful results.

Table 7.2 illustrates the properties of goal **G1**. This goal ensures the *Quality* of the investigation process. Furthermore, we derive the issue *Quality* and the object *Investigation Process* introduced in Section 4.2. The case study investigators are most interested in a qualitative process because they have to work with it. Thus, the viewpoint is taken from the *Investigator's* perspective.

Table 7.2.: Goal **G1** specification.

| | |
|---|---|
| Purpose | Quality Assurance |
| Issue | Quality |
| Object | Investigation Process (Process) |
| Viewpoint | Investigator |

To evaluate goal **G1**, we define the following questions and metrics to answer these questions:

#### Q1.1 – Profound Process

A process based on a theoretical basis and established knowledge (**R3**) has a strong foundation and considers crucial aspects (**Q1.1**). The process must synthesize an established process to conduct case studies and topic-specific aspects. For this thesis, topic-specific aspects are mobility-specific aspects and uncertainty propagation-specific processes.

**Q1.1** Is the introduced process profound?

**M1.1** Chapter 4 introduces the investigation process and refers further to the respective theoretical basis and established knowledge. We answer **Q1.1** argumentatively by discussing whether the referenced fundamentals adequately support the process definitions.

#### Q1.2 – Feasible Process

Furthermore, researchers need to be able to infer results (**Q1.3**) when following the proposed process. Trivially, a process that is not feasible is not a process. **R2** reflects not only on the case study but also on the case study investigation process (**Q1.2**).

**Q1.2** Is the introduced process feasible?

**M1.2** We answer question **Q1.2** argumentatively by applying the proposed investigation process and discussing, according to our experience afterward, whether we could infer any results.

#### Q1.3 – Meaningful Results

This question addresses furthermore that the proposed process must not only enable researchers to produce case study results but also these results must be expressive. This expressiveness is essential when analyzing the outcome and reporting to stakeholders.

**Q1.3** Does the process produce expressive case study results?

**M1.3** By applying the process, we infer results. Chapter 6 presents these results. We answer **Q1.3** argumentatively by looking at these results and discussing whether they answer the respective research question.

#### Note

We evaluate the remaining requirements within goal G2 to reduce evaluation redundancy. These requirements are *Replicability* (**R5**), *Comprehensibility* (**R4**), and *Theoretical Basis* (**R3**).

### 7.1.2. G2 – Qualitative Case Study

Besides the investigation process, quality is essential for the case selection and the case study design. the case must be *embedded in a real-world context* (**R1**). The other case study-specific requirements refer to the case study design. These requirements are *Feasibility* (**R2**), *Theoretical Basis* (**R3**), *Comprehensibility* (**R4**), *Replicability* (**R5**), and *Legal, Ethical, and Professional Requirements* (**R6**). According to Runeson et al., a good case study design adds to the quality of the respective case study [23].

Table 7.2 illustrates the properties of goal **G2**. The goal properties purpose, issue, and viewpoint follow the identical argumentation as for goal **G1**. However, the object of this goal is the *Case Study* and, transitively, the case study design, as mentioned before.

To evaluate goal **G2**, we define the following questions and metrics to answer these questions:

Table 7.3.: Goal **G2** specification.

| | |
|---|---|
| Purpose | Quality Assurance |
| Issue | Quality |
| Object | Case Study (Product) |
| Viewpoint | Investigator |

#### Q2.1 – Real-World Context

As already described in **R1** Section 5.3.1, the Uncertainty Impact Analysis (UIA) must be evaluated if it fulfills its goals in a real-life setting. Therefore, the modeled architecture must be near to reality.

**Q2.1** How close to reality is the case?

**M2.1** For the evaluation of **Q2.1**, we define a gradation from an unrealistic to a realistic case. Figure 7.1 illustrates the case realism scale. An unrealistic case is a trivial case because trivial problems should not require any software-based solution. The following gradation is a case not based on any shared understanding (e.g., not considering common software engineering concepts) while still representing a non-trivial case. For example, a model consisting of a single component accommodating the whole complexity of a problem is theoretically possible to build such a solution; however, it is not practicable. Closer to the real world is an architecture based on specifications, standards, and other artifacts commonly used in real-world software solutions. The last and most realistic step is an architecture derived from software that exists and is used in the real world.

Section 6.3 describes the model and references the used theoretical basis. According to this information, we can classify our case on the scale proposed in the previous paragraph and infer the degree of realism.

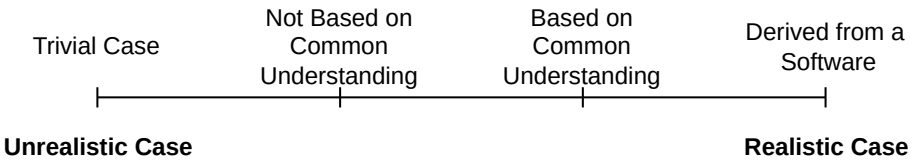


Figure 7.1.: Case gradation from unrealistic case to realistic case.

#### Q2.2 – Case Study Strategy Feasibility

Similar to **Q1.2** the case study strategy must also be feasible (**Q2.2**). This question goes beyond the investigation process's feasibility and refers to the entire case study design. According to Runeson et al. [23], a case study is feasible if

1. it considers the contemporaneous phenomenon in its real-life context,
2. the type of research questions is considered in the case study strategy, and

3. the degree of control is less critical.

This list represents the feasibility requirements.

**Q2.2** Is the case study strategy feasible?

**M2.2** The first feasibility requirement consists of two aspects real-life context and contemporaneousness. The first correlates with **Q1.1**. The real-life context is given if "yes" is the answer to **Q1.1**. Contemporaneousness is essential for being able to collect the required data. Therefore, we must look at the architectural model and if we can propagate uncertainties and the collect required results in the here and now.

For the second feasibility requirement, we must check if the type of research questions defined in Section 5.4 corresponds to the classification of the case study. Verner et al. state that questions "how" and "why" are appropriate for an exploratory case study [24]. Furthermore, they state that a case study can be evaluatory. However, they do not propose any question types for this type. In this thesis, the evaluatory part is the validation of the UIA. According to Oxford Learner's Dictionaries, *to validate* something is "to prove that something is true" [18]. Therefore, the answers to validating questions must be true or false, e.g., "Does", "Is", "Can". The case study fulfills this requirement if the research questions match the case study's classification defined in Section 5.5.

According to Runeson et al., p. 15, there is a trade-off between the degree of control and the degree of realism [23]. Further, the degree of control is low if analyzing qualitative data or if the case study design is flexible. **Q2.1** gives us the answer to the realism of the case. The data collection and analysis design, defined in Section 5.7, provides information about the data type. A design is flexible if the "key parameters of the study may be changed during the course of the study" [23]. This is the case if the investigation process enables investigators to adjust key parameters, e.g., architecture and research question.

We evaluate the fulfillment of each feasibility requirement argumentatively. We assume the case study strategy is feasible if Equation (7.1) applies.

$$1 = \frac{\text{number of fulfilled feasibility requirements}}{\text{total number of feasibility requirements}} \tag{7.1}$$

#### Q2.3 – Case Study Replicability

As **R5** reasons, enabling researchers to replicate the case study "should add to the validity of the research findings" [23]. Thus, a replicable case study will contribute to the quality of such. Because our case study deals with qualitative data, our understanding of *replicability* is less strict comparing to the replicability of an experiment. We understand replication rather as literal or theoretical replication [23, p. 16].

The requirement **R5** induces this question (**Q2.3**). The replicability of the case study is achievable through extensive planning and documentation (transparent). Requirements **R3** and **R4** induce this transparency. Therefore, we introduce two metrics: **M2.3** covers the theoretical basis (**R3**), whereas **M2.4** covers the comprehensibility (**R4**) of the case study strategy.

**Q2.3** Is the case study replicable?

**M2.3** Trivially, the first required domain of knowledge is about the research method case study. According to Runeson et al., a good case study must add to "existing knowledge by being based on previously established theory" [23]. Therefore, we elicit the required knowledge for our case study. Due to the defined rationales and objectives in Section 5.2, we require knowledge about uncertainties, their propagation, and their impact on confidentiality. Benkler covers structural uncertainty propagation [4]. However, he notes that the UIA might profit from dataflow uncertainty propagation. Thus, Benkler's work and these two propagation types outline further domain knowledge. The model is a component-based software architecture; therefore, we require Component-Based Software Engineering (CBSE) knowledge. Finally, the case lives in the mobility domain.

Required knowledge domains:

- Case study conduction
- Uncertainties in software architecture
- Structural uncertainty propagation through software architecture
- Uncertainty propagation along dataflow
- Confidentiality regarding uncertainties
- Knowledge about Benkler's work
- Palladio and CBSE
- Mobility domain

We count eight required knowledge domains. The *number of considered knowledge domains* defines how many of these domains were considered during the case study design and investigation. Therefore, we look at the case study design decisions and their explanations, presented in Chapters 4, 5, and 6. We count a knowledge domain as a *considered knowledge domain* if any of these chapters address such a knowledge domain by a reference. References referring to according foundations (Chapter 2) are also allowed. Equation (7.2) corresponds to metric **M2.3**. A value of 1 is desired and means that all required knowledge domains were considered.

$$\frac{\text{number of considered knowledge domains}}{\text{total number of knowledge domains}} \tag{7.2}$$

**M2.4** The requirement **R4** indicates that all decisions to be resolved for a *good case study* must be justified.

The construction of this metric is similar to the metric **M2.3**. First, we elicit the decisions to resolve according to case study guidelines by Runeson et al. [23] and case study protocol [8]. The list below presents elicited decisions. The elicited decision can be subdivided further. We derived decisions as broad as possible but as detailed as necessary. Additionally, we want to highlight that decisions regarding data collection and analysis were elicited from the importance of a clear chain of evidence mentioned by Runeson et al. [23].

- The rationale of the study
- Objectives of the study
- Classification of the required case study
- Requirements (each requirement)

- Case selection
- Architecture model
- Data collection (each orthogonal data set)
- Analysis (per each research question)
- Study limitations
- Threats to validity

The structure of this thesis enables tracking the case study decision. In the according section, we investigate if the decisions made were sufficiently justified. If so, we increase the *number of explained decisions* by 1.

Equation (7.3) corresponds to metric **M2.4**. A value of 1 is desired and means that all decisions were justified.

$$\frac{\text{number of explained decisions}}{\text{number of decisions}} \tag{7.3}$$

#### Q2.4 – Legal, Ethical, and Professional Requirements

As stated in **R6**, the collected data and the executed processes during the case study conduction must not defy legal, ethical, and professional requirements [23, p. 40] (**Q2.4**).

**Q2.4** Does the case study fulfill legal, ethical, and professional requirements?

**M2.5** Runeson et al. elaborate on legal, ethical, and professional requirements [23, pp.40-45]. We select the most critical aspects to our case study from their proposed aspects. We list the selected aspects below. **Q2.4** is answered with "yes", if …

… the researcher considered the collected data, produced outcome, and processing in the light of the General Data Protection Regulation (GDPR),

… subjects and organizations participating in the case study explicitly agreed to participate,

… confidential data is not needed for the case study; otherwise, if investigators met confidentiality agreements.

Runeson et al. state that the legal, ethical, and professional requirements must be considered already during the design phase. Therefore, we must examine Chapter 5 for the listed aspects and if the case study authors considered these aspects. We evaluate each aspect argumentatively and apply Equation (7.4) to answer **Q2.4**. We assume this question is answered if the following applies:

$$1 = \frac{\text{number of considered aspects}}{\text{total number of aspects}} \tag{7.4}$$

### 7.1.3. G3 – Case Study Comprehensiveness

Authorities commission investigators to execute a case study. In trivial cases, the authorities can be the investigators. The authorities provide their point of interest as requirements and questions. The case study must investigate all essential points of interest. Requirements **R10**-**R17** manifest our essential points of interest.

Table 7.4 gives an orientation of goal **G3**. The purpose of a case study is to investigate the phenomenon and to provide answers regarding this phenomenon. Trivially, the case study must provide answers to all questions. Therefore, the purpose of this goal is *Comprehensiveness Assurance*, and the issue is *Comprehensiveness*. The object of goal is the conducted *Case Study*. The main interest of a comprehensive case study lies with the authorities. However, the case study might also have other stakeholders interested in the comprehensiveness of the case study. We generalize such stakeholders as *Case Study Consumers*, which defines the viewpoint of this goal.

Table 7.4.: Goal **G3** specification.

| | |
|---|---|
| Purpose | Comprehensiveness Assurance |
| Issue | Comprehensiveness |
| Object | Case Study |
| Viewpoint | Case Study Consumer |

To evaluate goal **G3**, we define the following question and a metric to answer these questions:

#### Q3 – Case Study Comprehensiveness

Section 5.3.2 presents the analysis-specific requirements (**R10**-**R17**). These requirements manifest the points of interest and the need to be resolved by the case study (**Q3**).

**Q3** Does the case study consider all analysis-specific requirements?

**M3** By design, we mapped these requirements to the research questions the case study must answer. Thus, we look at Chapter 6, which discusses the case study results, and check if this chapter provides answers to defined research questions. We formulate this approach as follows:

Let $RQ$ be a set of defined research questions and $A$ the set of answers to the research questions. Let further $\psi : RQ \rightarrow A$ be a mapping that maps research questions to answers. $\psi^{-1}(A)$ is the set of all answered research questions.

Now, from the definitions of the research questions (Section 5.4), we can trace back to the individual requirements because the requirements are mapped to research questions by design.

Let $\Phi$ be a mapping from domain $R$ (analysis-specific requirements) to co-domain $RQ$ (research questions). $\Phi^{-1}(RQ)$ is the set of all requirements mapped to a research question from $RQ$. The case study is fully comprehensive if $Comprehensiveness = 1$ according to Equation (7.5).

$$Comprehensiveness = \frac{|\Phi^{-1}(\psi^{-1}(A))|}{|R|} \tag{7.5}$$

## 7.2. Evaluation Results

In this section, we execute the evaluation and discuss the results according to the proposed design, Section 7.1. Each goal is structured as follows: First, we summarize the goal, then we elaborate on each question, and lastly, we discuss the results of each question.

### 7.2.1. G1 – Qualitative Investigation Process

The proposed process must guide the researcher through the case study and contribute to comprehensive results of good quality. In Section 7.1.1, we demand the process to be profound, feasible and produce meaningful results.

#### Q1.1 – Profound Process

We argue that the introduced process is profound (**Q1.1**) because we aligned it with well-known and established literature. We used the book *Case Study Research in Software Engineering: Guidelines and Examples* by Runeson et al. [23]. These guidelines gave the process the fundamental structure. They highlight that a case study protocol is the main guideline document for investigators. Therefore, we implemented the suggested protocol structure by Brereton et al. [8] to support our process. Figure 7.2 depicts all proposed protocol elements and where we elaborated on these elements during this thesis.

As presented in Figure 7.2, we elaborated on most of the protocol elements during the design phase in Chapter 5. The *Background* reflects additionally onto the thesis's introduction. Section 7.4 presents, amongst others, the *Study Limitations*. Appendix A represents the protocol element *Appendices*. More interesting are the remaining protocol elements. We implicitly consider the *Plan Validity* element in this chapter. Before the thesis, a proposal was elaborated, including a case study schedule. We present the proposal and the thesis in the scope of this bachelor's thesis. Both presentations constitute the *Reporting* element. As this figure states, no protocol elements were omitted.

The evidence collection step, proposed by Runeson et al. [23], is subdivided into steps **/3/** to **/7/**. During these steps, the evidence is collected by uncertainty types derivation, model creation, uncertainty annotations, and execution of the UIA. This decision follows the proposed uncertainty type derivation process and the UIA approach proposed by Benkler [4].

The justification of the defined roles follows the elaborations of Runeson et al. [23] and Benkler [4].

#### Q1.2 – Feasible Process

According to metric **M1.2**, we applied the proposed case study investigation process to conduct a mobility case study. Chapter 6 shows that we can execute the process and

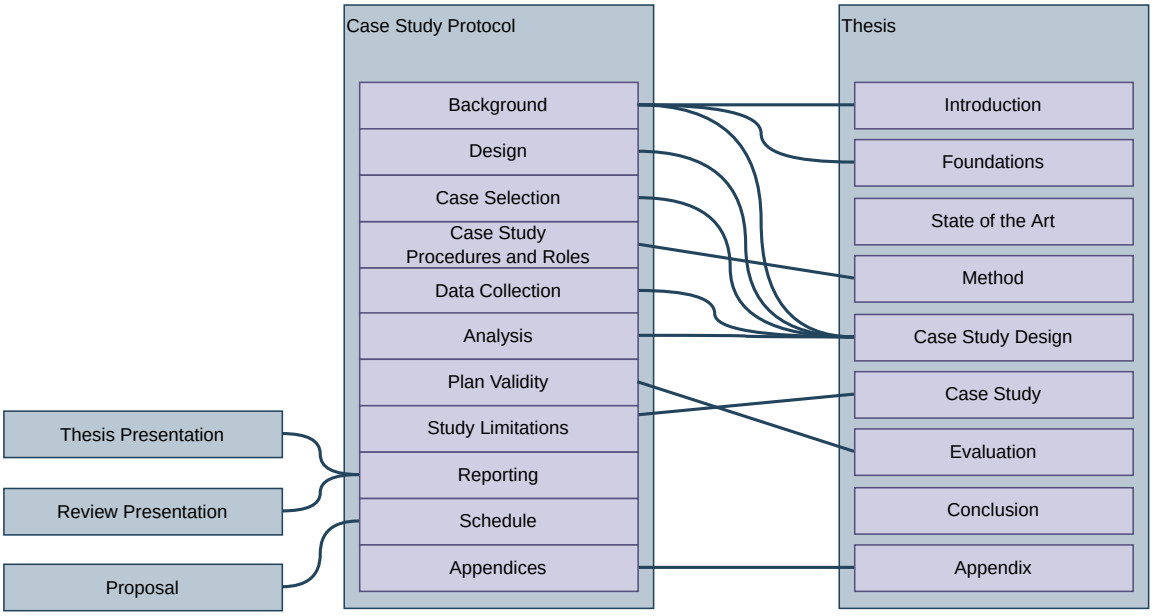


Figure 7.2.: Mapping case study protocol elements to thesis artifacts and thesis outline.

produce results. We stored the created model, the collected data, and the results according to Section 7.5.

We believe our process is feasible (**Q1.2**) because we successfully conducted a mobility case study and produced results (Chapter 6).

#### Q1.3 – Meaningful Process

According to metric **M1.3**, we applied the proposed case study investigation process to conduct a mobility case study. We believe the produced results are meaningful (**Q1.3**) because they point out the weaknesses and strengths of the UIA, as well as give insides into the mobility domain regarding uncertainties.

#### Result

In summary, we believe the proposed investigation process is profound, feasible, and produces meaningful results. Thus, goal **G1** is fulfilled, and so the process is a qualitative investigation process.

### 7.2.2. G2 – Qualitative Case Study

In this section, we evaluate the quality of the case study. In order to do this, we answer the questions, defined in Section 7.1.2, about the case proximity to reality (**Q2.1**), the feasibility of the case study strategy (**Q2.2**), the replicability of the case study (**Q2.3**), and finally, if the researcher considered legal, ethical, and professional requirements (**Q2.4**).

#### Q2.1 – Real-World Context

To answer **Q2.1** about the proximity of the modeled architecture to a real-life project, we use the metric **M2.1**. We used Mobility Data Specification (MDS) to model the architecture. This specification is well-known and well-used. According to MDS-Website[1] more than 40 cities and more than 40 providers use MDS. Furthermore, as we described in Section 6.3.2, we picked two use cases used in the real world [16]. Further, the System exposes three additional interfaces *IPublicClient*, *IMobileUser*, and *IVehicleConnector* that are motivated by the MDS. We argue that our model is close to reality, because it is based on common understanding, specifications, and the use cases are used in the real world.

#### Q2.2 – Case Study Strategy Feasibility

Metric **M2.2** suggests to check first the contemporaneousness of the phenomenon, then the type of the research questions, and lastly, the required degree of control.

We are investigating a contemporaneous phenomenon because the model exists, and we can derive uncertainty types and execute the UIA in the here and now whenever we want to collect the required data.

Regarding the research question types: Section 5.5 classifies our case study as (1) exploratory and (2) evaluatory. The only research question not following the questions schema defined by metric **M2.2** is **RQ2**. However, the answer to question RQ2, which uncertainties exist in the mobility domain, will provide additional or new insides into the mobility domain. Hence, **RQ2** is an exploratory question.

The degree of control is low because …

- … the case is close to reality, as stated previously (**Q2.1**).
- … some collected data are qualitative, as declared in Section 5.7.
- … the strategy is flexible due to the iterative and incremental design of the case study. This flexibility enables researchers to adjust key parameters of the case study. Section 4.2 describes such adjustments during process steps /**1**/ and /**2**/.

All three feasibility requirements are fulfilled, and according to Equation (7.1), we achieve the desired value $3/3 = 1$. Thus, we consider our case study strategy feasible (**Q2.2**).

#### Q2.3 – Case Study Replicability

To evaluate the replicability of our case study (**Q2.3**), we will use metrics **M2.3** and **M2.4**. Metric **M2.3** evaluates the applied theoretical basis. In several sections of this thesis, we provided the required theoretical basis. Table 7.5 references the required knowledge. We do not claim the completeness of the presented references in this table. Instead, the table contains examples; this is sufficient to answer this question. We considered this knowledge during the case study design and investigation by design. According to Equation (7.2), we achieve the desired value $8/8 = 1$.

Metric **M2.4** evaluates the comprehensibility of the case study. Table 7.6 maps decisions made to the respective justification. We consider these justifications satisfactory. This

[1] `https://www.openmobilityfoundation.org/mds-users/`

Table 7.5.: References to theoretical basis.

| Theoretical Basis | References |
|---|---|
| Case study conduction | Chapters 4 and 5 |
| Uncertainties in software architecture | Sections 2.4, 5.7.1.2, and 5.7.1.3 |
| Structural uncertainty propagation | Section 5.7.1.2 |
| Uncertainty propagation along dataflow | Section 5.7.1.3 |
| Confidentiality regarding uncertainties | Section 5.7 |
| Knowledge about Benkler's work | Chapters 4, 5, and 6 |
| CBSE | Section 6.3 |
| Mobility Domain | Section 2.2 |

mapping, in conjunction with Equation (7.3) ($10/10 = 1$), confirms that our case study is comprehensible.

Table 7.6.: References to justifications of decisions made.

| Decisions | References to Justifications |
|---|---|
| The rationale of the study | 5.2 |
| Objectives of the study | 5.2 |
| Classification of the required case study | 5.5 |
| Requirements (each requirement) | 5.3 |
| Case selection | 6.2 |
| Architecture model | 6.3 |
| Data collection (each orthogonal data set) | 5.7 |
| Analysis (per each research question) | 5.7 |
| Study limitations | 7.4 |
| Threats to validity | 7.3 |

In summary, our case study has a solid theoretical basis and is comprehensive. Thus, our case study is replicable (Q2.3).

#### Q2.4 – Legal, Ethical, and Professional Requirements

Section 5.8 describes the respective considerations. We check if all aspects defined by metric **M2.5** were considered. Case study authors considered the collected data, produced outcomes, and executed processes in light of the GDPR. Furthermore, the case study authors have given thoughts about non-disclosure agreements and informed consent. According to the metric **M2.5**, the desired value was achieved: Equation (7.4): $3/3 = 1$.

The conducted case study does not defy legal, ethical, and professional requirements because they were already considered during the case study design.

#### Result

According to the evaluation of questions **Q2.1** to **Q2.4**, goal **G2** is achieved. However, there are certain threats to validity. Section 7.3 explains these threats and describes how we handled them.

### 7.2.3. G3 – Case Study Comprehensiveness

The last goal evaluates the comprehensiveness of the case study by considering the analysis-specific requirements defined in Section 5.3.2 and the research questions defined in Section 5.4. This goal is essential for the satisfaction of the case study consumers.

#### Q3 – Case Study Comprehensiveness

Overall, we introduced nine analysis-specific requirements. During the design phase of the case study, we derived research questions from these requirements. We omitted requirements **R10** and **R11** because we considered covering seven of nine requirements in the first iteration sufficient. Unfortunately, due to time constraints, we could not execute any further iterations and derive further research questions. However, the case study answers all the derived research questions in Chapter 6. According to metric **M3**, we achieve a value of $7/9 = 0.7778$, meaning we cover 77.78% of the requirements.

#### Result

The evaluation of **Q3** shows that the conducted case study is not complete; however, it is very comprehensive, reaching about 77.78% coverage of analysis-specific requirements. Thus, we consider goal **G3** not completely fulfilled To reach a coverage of 100 % further iterations are necessary.

## 7.3. Threats to Validity

This section lists identified threats to validity and how we handled them. Runeson et al. classified validity into four categories: (1) internal validity, (4) external validity, (3) construct validity, and (4) reliability [23, p. 71]. We categorize the threats according to this validity classification.

### Threats to Internal Validity

Runeson et al. emphasize that bias seriously threatens internal validity when conducting a case study [23, p. 4]. In our case, the bias emerges because the author validates his work. The proper research methodology was applied to mitigate the threat, as Runeson et al. suggested. Additionally, external parties reviewed the approach.

The author has little experience regarding uncertainty propagation and impact assessment on confidentiality. Such experience is valuable when deriving the *Affected Set* used as a gold standard during the case study analysis. This setting threatened internal validity and was handled by consulting experts.

### Threats to External Validity

The case study investigates the mobility domain. Thus, we created an architectural model which however does not represent the entire mobility domain. This circumstance threatens external validity, which we handled by consulting the MDS.

### Threats to Construct Validity

Threats to construct validity arise when a researcher does not investigate precisely what has been defined as a point of interest in the case study. To mitigate this threat, we used a top-down approach for case study analysis and evaluation of our work. For the evaluation, we used the GQM approach. For the case study analysis, we used a similar approach. First, we defined requirements. Then, we derived research questions and defined respective analysis strategies based on these requirements.

### Threats to Reliability

The data and the analysis might be affected by the evaluating researcher. To handle this threat, we applied the GQM approach and, as mentioned in Section 7.3, consulted experts and executed reviews. Furthermore, we documented the evaluation design in Section 7.1 and published all artifacts created during this thesis to enable other researchers to re-evaluate our approach.

## 7.4. Assumptions and Limitations

In this section, we present limitations, and which and why we made assumptions, as well as the justification why these assumptions can be made.

The foundation of this thesis is the book *Case Study Research in Software Engineering : Guidelines and Examples* by Runeson et al. [23]. We assume that these guidelines will ensure a comprehensive case study, accurate data collection and analysis, and a well-documented case study.

The *Uncertainty Type Derivation Process proposed by Benkler [4] is part of the case study. We assume that the process is accurate and use it without further validation, as additional validation would exceed the scope of this thesis. However, Benkler conducted a case study to validate the *Uncertainty Type Derivation Process* and achieved good results.

To our knowledge, there does not exist any definition of uncertainty propagation alongside the data flow. Therefore, we describe a possible propagation. We can not evaluate this solution because this is out of the scope of this thesis. We assume this propagation is sufficiently sound because the derived *Affected Sets* were discussed with experts that supervise this thesis and these sets were fine enough.

The central point of the case studies is the UIA itself. Benkler highlights some functional limitations of the Uncertainty Impact Analysis. He states that “not all the necessary propagation algorithms are implemented yet” [4]. This reflects our limitations because we could not propagate several uncertainties that require missing propagation algorithms and explore the mobility domain further.

## 7.5. Data Availability

The case study-specific data, such as the architectural model, case scenarios, and collected data, are publicly available [19].

# 8. Conclusion

In this chapter, we summarize our work and our results in Section 8.1. Section 8.2 provides an outlook.

## 8.1. Summary

In this thesis, we determined a gap in the state of the art and tried to close it by introducing a framework for a particular class of case studies. Such case studies validate an Uncertainty Impact Analysis (UIA) regarding confidentiality. We created the framework on a sound foundation that comprises an investigation process and a case study protocol. We defined roles and procedures for the investigation process. This way, we improved clarity and specialized the general case study conduction process to our demands. We composed a case study protocol that should guide researchers through the entire investigation process.

To evaluate our framework, we conducted a mobility case study by using our framework. For the study, we defined a case from Mobility Data Specification (MDS) and modeled it as a component-based software architecture by using Palladio Component Model (PCM). We then examined the mobility domain to find uncertainty types and used them to annotate the model with uncertainties and propagate these throughout the architecture by using the UIA. Additionally, manual propagation was performed. The analysis of the collected data produced meaningful results. Our results imply that structural propagation is insufficient to provide software architects with a reliable starting point for further confidentiality analyses.

For the evaluation, we set a Goal-Question-Metric (GQM) plan. The evaluation subjects were the investigation process and the case study protocol. Our results imply that our framework has a sound basis, is feasible, and promotes meaningful results.

This framework aims to support researchers validating UIA regarding confidentiality. Furthermore, the conducted mobility case study validates Benkler's UIA and implies that Benkler's assumption that data flow propagation could enhance the UIA might be valid.

## 8.2. Future Work

As mentioned in the previous section, the proposed framework consists of two components the case study investigation process and the case study protocol. The evaluation showed that both components are profound, transparent, and comprehensive. Furthermore, this framework contributes to meaningful results. However, we also discovered weaknesses in our approach, e.g., incomplete implementation of the defined requirements. Nevertheless, we see potential in this framework for future case studies. Therefore, we suggest several improvements.

**Further Iterations** In the scope of this thesis, we executed one iteration. Our case and scenario could lead us to some conclusions. However, both are still rudimental and could profit from further iterations. During further iterations researchers could implement remaining analysis-specific requirements and, thus, increase the study's comprehensiveness. In future iterations, the quality assurance might be improved by considering checklists for case study phases, e.g., Runeson et al. composed such checklists for general case study research in software engineering.

**Uncertainty Propagation Alongside the Data Flow** When this thesis is elaborated, we could not find any profound definition of how uncertainties propagate alongside the data flow. More literature on this will probably be published soon. Consulting this literature can improve the manual estimation of the impact and affected set. Thus, improve the reliability of future case studies conducted according to our framework.

**Generalization** Our framework is geared towards the mobility domain and the UIA proposed by Benkler. The more accessible part is to generalize the domain because, in contrast to the process, the design is dependent on the domain. A slightly more complicated part is the generalization regarding the validated UIA.

**User Study** Especially in the case of generalization, it might be exciting to execute a user study. A user study could contribute to better framework reliability and determine possible refinements. The setting of a user study could be: Ten students or researchers conduct case studies according to the framework and document their experience, which is evaluated afterward. The case studies might be conducted in different domains. This approach could further extend the knowledge of uncertainties, their propagation, and their impact on confidentiality.

# Bibliography


[1] Victor R Basili, Gianluigi Caldiera, and H Dieter Rombach. "The Goal Question Metric Approach". In: (1994), pp. 528–532.

[2] Victor R. Basili and David M. Weiss. "A Methodology for Collecting Valid Software Engineering Data". In: *IEEE Transactions on Software Engineering* SE-10.6 (1984), pp. 728–738. DOI: 10.1109/TSE.1984.5010301.

[3] Steffen Becker, Heiko Koziolek, and Ralf Reussner. "The Palladio Component Model for Model-Driven Performance Prediction". In: *Journal of Systems and Software* 82.1 (2009), pp. 3–22. ISSN: 0164-1212. DOI: 10.1016/j.jss.2008.03.066. URL: https://www.sciencedirect.com/science/article/pii/S0164121208001015.

[4] Niko Benkler. "Architecture-Based Uncertainty Impact Analysis for Confidentiality". Karlsruher Institut für Technologie (KIT), 2022. DOI: 10.5445/IR/1000144641.

[5] Niko Benkler. *Architecture-Based Uncertainty Impact Analysis for Confidentiality (Reproduction Set)*. Zenodo, Feb. 2022. DOI: 10.5281/zenodo.6202288. URL: https://doi.org/10.5281/zenodo.6202288.

[6] Niko Benkler. *Uncertainty Impact Analysis (UIA)*. FluidTrust, Jan. 8, 2022. URL: https://github.com/FluidTrust/Palladio-Addons-Uncertainty-ImpactAnalysis (visited on 06/02/2022).

[7] B. Boehm and V.R. Basili. "Top 10 List [Software Development]". In: *Computer* 34.1 (2001), pp. 135–137. DOI: 10.1109/2.962984.

[8] Pearl Brereton et al. "Using a Protocol Template for Case Study Planning". In: *12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12*. 2008, pp. 1–8.

[9] S. Hahner. "Architectural Access Control Policy Refinement and Verification under Uncertainty". In: *CEUR Workshop Proceedings*. Vol. 2978. 2021. URL: www.scopus.com.

[10] Sebastian Herold et al. "CoCoME - The Common Component Modeling Example". In: *The Common Component Modeling Example*. Ed. by Andreas Rausch et al. Vol. 5153. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 16–53. ISBN: 978-3-540-85288-9 978-3-540-85289-6. DOI: 10.1007/978-3-540-85289-6_3. URL: http://link.springer.com/10.1007/978-3-540-85289-6_3 (visited on 06/11/2022).

[11] International Organization for Standardization. *ISO/IEC 27000:2018(En), Information Technology — Security Techniques — Information Security Management Systems — Overview and Vocabulary*. URL: https://www.iso.org/obp/ui/#iso:std:iso-iec:27000:ed-5:v1:en (visited on 10/14/2022).

[12] Klaus Krogmann and Ralf Reussner. "Palladio – Prediction of Performance Properties". In: *The Common Component Modeling Example: Comparing Software Component Models*. Ed. by Andreas Rausch et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 297–326. ISBN: 978-3-540-85289-6. DOI: 10.1007/978-3-540-85289-6_12. URL: https://doi.org/10.1007/978-3-540-85289-6_12.

[13] Open Mobility Foundation (OMF). *About MDS | Open Mobility Foundation*. 2020. URL: https://www.openmobilityfoundation.org/about-mds/ (visited on 05/25/2022).

[14] Open Mobility Foundation (OMF). *GitHub Repository: Mobility Data Specification*. GitHub repository. 2021. URL: https://github.com/openmobilityfoundation/mobility-data-specification (visited on 10/19/2022).

[15] Open Mobility Foundation (OMF). *Mobility Data Specification – Privacy Guide for Cities*. 2020. URL: https://raw.githubusercontent.com/openmobilityfoundation/governance/main/documents/OMF-MDS-Privacy-Guide-for-Cities.pdf (visited on 10/19/2022).

[16] Open Mobility Foundation (OMF). *Use Case Gallery*. Airtable. URL: https://airtable.com/shrPf4QvORkjZmHIs/tblzFfU6fxQm5Sdhm (visited on 10/17/2022).

[17] Open Mobility Foundation (OMF). *Using MDS Under GDPR | Open Mobility Foundation*. Nov. 18, 2021. URL: https://www.openmobilityfoundation.org/using-mds-under-gdpr/ (visited on 10/17/2022).

[18] Oxford University Press. *Oxford Learner's Dictionaries | Find Definitions, Translations, and Grammar Explanations at Oxford Learner's Dictionaries*. URL: https://www.oxfordlearnersdictionaries.com/ (visited on 10/12/2022).

[19] Denis Priss. "A Mobility Case Study Framework for Validating Uncertainty Impact Analyses Regarding Confidentiality". Karlsruhe Institute of Technology, Oct. 2022. DOI: 10.5281/zenodo.7236311. URL: https://doi.org/10.5281/zenodo.7236311.

[20] Christoph Rathfelder and Benjamin Klatt. "Palladio Workbench: A Quality-Prediction Tool for Component-Based Architectures". In: *2011 Ninth Working IEEE/IFIP Conference on Software Architecture*. 2011, pp. 347–350. DOI: 10.1109/WICSA.2011.55.

[21] Ralf Reussner et al., eds. *Modeling and Simulating Software Architectures: The Palladio Approach*. Cambridge, Massachusetts: The MIT Press, 2016. xiv, 377 Seiten : Illustrationen, Diagramme ; 24 cm. ISBN: 978-0-262-03476-0.

[22] Kiana Rostami et al. "Architecture-Based Assessment and Planning of Change Requests". In: *Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures*. 2015, pp. 21–30.

[23] Per Runeson et al. *Case Study Research in Software Engineering: Guidelines and Examples*. Hoboken, UNITED STATES: John Wiley & Sons, Incorporated, 2012. ISBN: 978-1-118-18102-7. URL: http://ebookcentral.proquest.com/lib/karlsruhetech/detail.action?docID=818522 (visited on 05/25/2022).

[24] J.M. Verner et al. "Guidelines for Industrially-Based Multiple Case Studies in Software Engineering". In: *2009 Third International Conference on Research Challenges in Information Science*. 2009, pp. 313–324. DOI: 10.1109/RCIS.2009.5089295.

[25] Warren E Walker et al. “Defining Uncertainty: A Conceptual Basis for Uncertainty Management in Model-Based Decision Support”. In: *Integrated assessment* 4.1 (2003), pp. 5–17.

[26] Claes Wohlin. “Case Study Research in Software Engineering—It Is a Case, and It Is a Study, but Is It a Case Study?” In: *Information and Software Technology* 133 (2021), p. 106514. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2021.106514. URL: https://www.sciencedirect.com/science/article/pii/S0950584921000033.

[27] Robert K Yin. *Case Study Research: Design and Methods*. Vol. 5. sage, 2009.

# A. Appendix

## A.1. Abbreviations

**ADD** Architectural Design Decision.

**API** Application Programming Interface.

**CBSE** Component-Based Software Engineering.

**CoCoME** Common Component Modeling Example.

**CWA** Corona Warn App.

**GDPR** General Data Protection Regulation.

**GQM** Goal-Question-Metric.

**KAMP** Karlsruhe Architectural Maintainability Prediction.

**MDS** Mobility Data Specification.

**PCM** Palladio Component Model.

**UIA** Uncertainty Impact Analysis.

## A.2. Model Diagrams

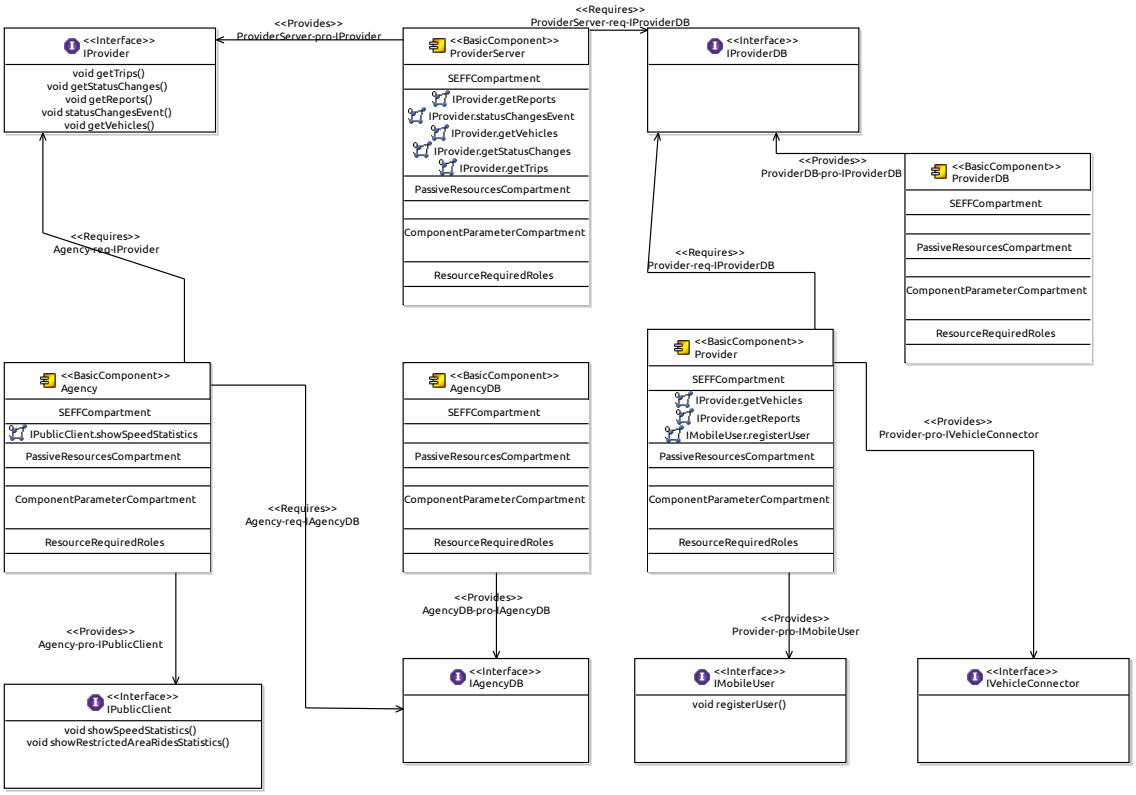


Figure A.1.: Repository diagram for the MDS-System.

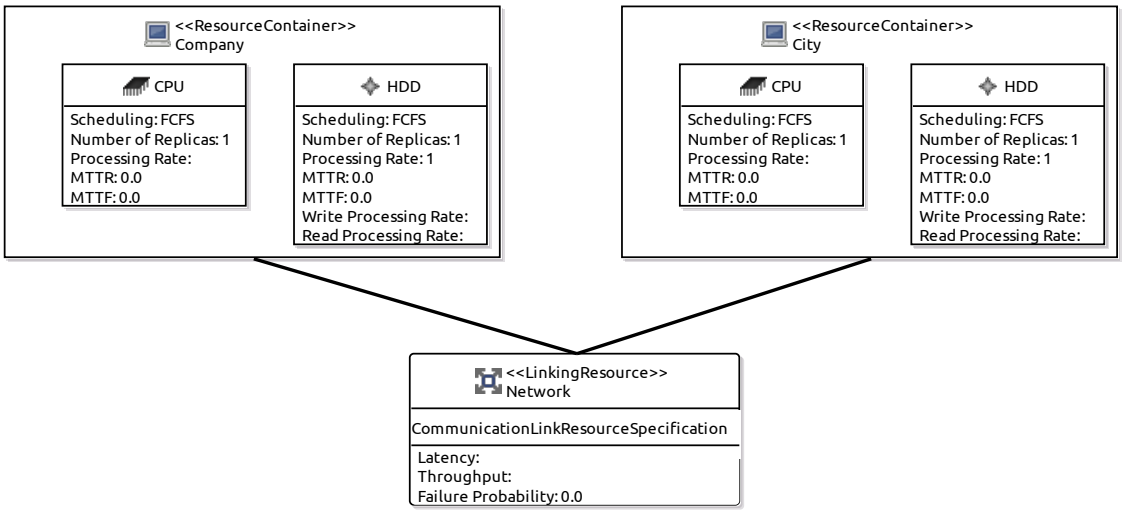


Figure A.2.: Resource environment diagram for the MDS-System.

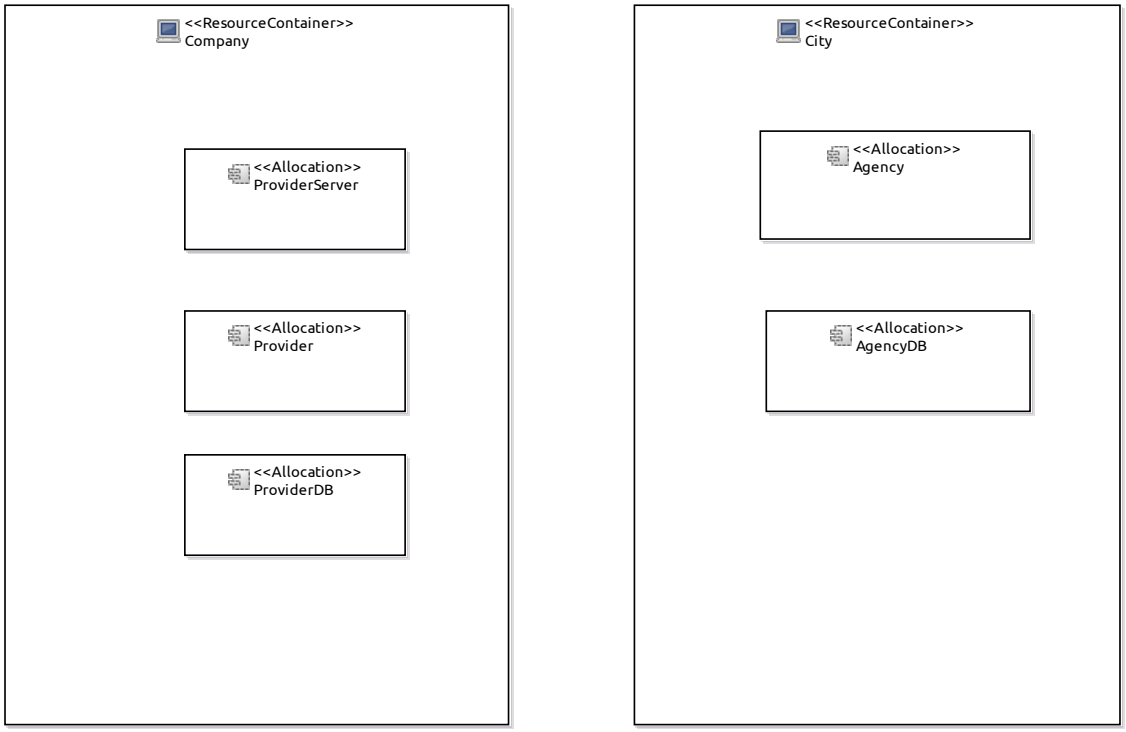


Figure A.3.: Allocation diagram for the MDS-System.

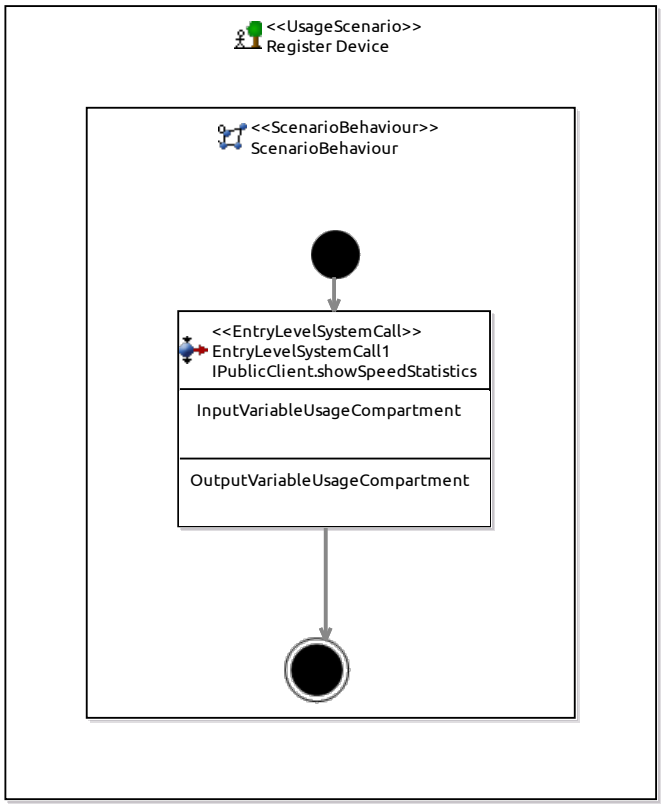


Figure A.4.: Usage model diagram for the MDS-System.