Modelling Data Protection in Fog Computing Systems using UMLsec and SysML-Sec

Jan Laufer, Zoltán Ádám Mann, Andreas Metzger
paluno – The Ruhr Institute for Software Technology, University of Duisburg-Essen, Essen, Germany
firstname.lastname@paluno.uni-due.de

***Abstract*—Fog computing provides low-latency cloud-like compute resources to end devices, thereby facilitating the delivery of modern data-intensive applications at the edge. These applications must comply with data protection requirements, such as posed by the European General Data Protection Regulation, which requires that protection of personal data must be ensured *by design*. We analyse to what extent UMLsec and SysML-Sec, extensions of the widely used modelling languages UML and SysML, help modelling data protection aspects during the design of fog computing systems. We use UMLsec and SysML-Sec because these languages are capable of modelling information security aspects, which significantly overlap with data protection aspects. As basis for our analysis, we create UMLSec und SysML-Sec models for three real-world use cases from Smart City, Smart Manufacturing, and Smart Media. Using real-world use cases facilitates reflecting actual data protection concerns in practice. The results indicate that both UMLsec and SysML-Sec are partially suitable for capturing the data protection aspects identified in the use cases. Based on the identified gaps, we propose potential enhancements of these languages.**

***Index Terms*—Fog computing, Edge computing, Data protection, Security, Privacy, UMLsec, SysML-Sec**

I. Introduction

Motivation: Many modern applications, such as those used in smart cities and smart industry, require low network communication latency combined with satisfactory quality of service for real-time data processing [1]. Since most end devices (e.g. IoT devices) lack the necessary resources for real-time data processing, offloading data storage and processing is a possible solution to this problem [2], [3]. Fog computing is designed to satisfy these requirements by providing computing capacity at the edge of the network in so-called fog nodes [2]. Fog nodes can perform certain tasks that the end devices are not capable of. Fog nodes can also pre-process and compress data, for example to reduce data load that may be sent to the cloud for further processing [1].

The characteristics of fog computing (e.g. mobility and location awareness of fog nodes) have given rise to new design challenges with regard to personal data protection and information security [4], [5]. Laws such as the General Data Protection Regulation (GDPR) of the European Union (EU) dictate the protection of personal data [6]. Data protection requirements and threats known from the areas of information security, Internet of Things (IoT), and cloud computing as well as newly emerging data protection threats must be considered in the scope of fog computing [7]. For instance, due to the dynamic network structure of fog computing systems, an unknown and untrusted fog node provider could access personal data that is routed to and processed in its fog node [8]. In the rest of the paper, data protection requirements and threats to data protection are referred to as data protection aspects.

Problem statement: Data protection aspects should be taken into account during system design. For example, the GDPR requires that data processing systems ensure data protection *by design*. UML[1] and SysML[2], well-known standardized modelling languages, are widely used for designing systems. The extensions UMLsec [9], [10] and SysML-Sec [11] foster the modelling of secure systems. We expect that fog computing systems can also be modelled with UMLsec and SysML-Sec. As explained in Sec. II, data protection has significant overlaps with information security. Thus, UMLsec and SysML-Sec may serve as a starting point to model data protection concerns. However, UMLsec and SysML-Sec were not conceived with the purpose of modelling data protection in fog computing systems. This is because information security and data protection are not equivalent, and fog computing systems have special characteristics.

Contribution: The aim of the paper is to analyse to what extent UMLsec and SysML-Sec can be used to model data protection aspects in fog computing systems and which gaps and thus needs for enhancement may exist. We conduct this examination on a use case basis, reflecting on decisions made during the modelling process in the evaluation. Three real-world use cases that employ fog computing in diverse application domains are used: Smart City, Smart Manufacturing and Smart Media. These use cases serve to validate research results in the EU-funded research project FogProtect [12]. We model data protection aspects of the use cases, and identify concrete modelling challenges and gaps, which are then discussed.

Work partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement no. 871525 (FogProtect). Useful discussions with project partners are gratefully acknowledged.

Paper published in ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), pp. 777-786, 2021.

[1] See https://www.omg.org/spec/UML/About-UML/

[2] See https://www.omg.org/spec/SysML/1.6/About-SysML/

II. Preliminaries

A. Data protection

Data protection refers to the protection of personal data, as stipulated by legal regulations. Data protection is related to information security but not the same. It is helpful to understand both the information security and the legal perspective.

ISO 27000:2018 describes information security as the "preservation of confidentiality, integrity and availability of information" [13]. Moreover, other information security aspects such as authenticity, accountability, non-repudiation, and reliability should also be considered.

From the perspective of the GDPR, data protection mechanisms should prevent personal data breaches [6]. Personal data means "any information relating to an identified or identifiable natural person", for example name, address or location data [6]. A personal data breach means "a breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorised disclosure of, or access to, personal data transmitted, stored or otherwise processed" [6].

Therefore, data protection includes information security aspects such as confidentiality or integrity concerning personal data. However, data protection goes beyond information security, for example by defining specific roles as well as their rights and obligations related to personal data. GDPR-based roles that appear in this work are the following:

Data Subject: A data subject is "an identified or identifiable natural person . . . who can be identified . . . by reference to an identifier such as a name . . . or to one or more factors specific to the . . . identity of that natural person" [6].

Data Controller: A data controller "determines the purpose and means of the processing of personal data" [6].

Data Processor: A data processor "processes personal data on behalf of the [data] controller" [6].

Third Party: A third party is authorised to process personal data under the direct authority of the controller or processor.

From a legal point of view, data protection must be ensured in a fog computing system processing personal data. In order to achieve data protection, information security aspects like the preservation of confidentiality need to be taken into account.

B. Modelling languages used

In this paper we concentrate on selected structure-oriented diagram types of UMLsec and SysML-Sec. They complement each other by representing different perspectives. For example, both languages enable the modelling of security properties of data transfer. For this perspective we decided to use UMLsec deployment and class diagrams instead of SysML-Sec block definition and internal block diagrams. This is because in SysML-Sec different security properties on data channels are illustrated by the same lock symbol and are visually indistinguishable. Next, the diagram types used are introduced. Concrete graphical examples can be found in Sec. V.

1) UMLsec: UMLsec is an extension of the Unified Modeling Language (UML) [9], [10], introducing new stereotypes, tags and conditions. Stereotypes and tags are used, for example, to define connections as encrypted («encrypted»), to link data with security requirements on the logical level (e.g., «secrecy» or «integrity») and to define information security guidelines for a system (e.g., «secure links»). A complete description of all stereotypes and tags can be found in [10]. By using condition checking it can be verified whether security requirements are met by the system design. The following diagram types are used in this work:

Deployment Diagrams are used to describe the physical structure of a system, the deployment of software components, and the interconnectivity within and between hardware elements. With UMLsec, the modelling elements node, component, link (physical communication link), and dependency (logical connection between components) may be annotated with stereotypes and tags. For example, to represent required communication security, the dependencies between components are annotated with tags. A link between two nodes may be tagged with a stereotype to describe the connection type. It is also possible to annotate components with stereotypes. By annotating a deployment diagram with the stereotype «secure links», the diagram can be mapped to a specific adversary type. An *adversary table* maps stereotypes to a set of actions (threats) that adversaries of a defined type are capable of.

Using the adversary table, it can be determined, for instance, whether the connection type between two nodes provides sufficient communication security for a dependency between two components that is annotated with a security requirement. If a dependency is annotated with the stereotype «high», the adversary table for the given connection type is not allowed to have any elements in the corresponding threat list. When using the stereotype «secrecy», the threat list in the adversary table for the given connection type must not include "read". When using «integrity», "insert" is not allowed to be part of the threat list of the given connection type.

In **Class Diagrams**, classes, properties and dependencies can be enriched with UMLsec stereotypes and tags. This allows both the presentation of information security requirements and the examination of whether these requirements are fulfilled. Annotating a class diagram with the stereotype «secure dependency» means that dependencies between classes respect the information security requirements on the data that may be communicated across them. Moreover, if two classes that are connected via a dependency share a property with the same name, information security requirements on the shared properties have to be consistent between the two classes. A class may be annotated with the stereotype «critical». Afterwards tags can be used to specify the information security requirements of properties which represent data. Dependencies between classes can also be enriched with stereotypes.

For example, if a property x of class A is tagged with {secrecy} and a dependency annotated with «call» or «send» between class A and class B exists, the dependency has to be stereotyped «secrecy» to fulfil the «secure dependency» requirements. Moreover, if class B also has a property with the same name as x, it also needs to be tagged with {secrecy}.

2) SysML-Sec: SysML-Sec is based on the Systems Modeling Language (SysML) [11]. SysML-Sec was designed to

consider security and safety in the early design and development phases in relation to software and hardware components. In order to complement the UMLsec diagrams in a meaningful way, we decided to use the following diagram types.

Requirement Diagrams represent a hierarchy of requirements that may be connected to each other with "derive" dependencies and "containment" relationships. SysML-Sec introduces the stereotype «Security Requirement». A security requirement can be categorised with the "kind" property and ranked with the "risk" property. Each security requirement can be linked to an attack from the parametric diagram.

Parametric Diagrams model attack trees. Each potential attack is tagged with the stereotype «attack» or «root attack». «root attack» means that this attack is the root of an attack tree. Each attack is part of a block representing the target of the attack. A block is visualized by a rectangle with a name. To link multiple attacks, logical and temporal operators can be used. Countermeasures are tagged with the stereotype «countermeasure» and linked to attacks that they negate.

III. Research Approach

As stated in Sec. I, we expect that UMLsec and SysML-Sec are suitable for modelling many aspects of fog computing systems. However, fog computing systems may have specific characteristics, which were not taken into account in the development of both languages. In addition, both languages focus on security aspects, which largely overlap with data protection, but are not the same. Thus, we pose the following research question.

RQ: *To what extent are UMLsec and SysML-Sec suitable for modelling data protection aspects in fog computing?*

This question entails two main concerns. First, it is to be expected that some limitations of the languages stem from special characteristics of fog computing systems. Therefore, we examine to what extent modelling of fog computing systems is possible with UMLsec and SysML-Sec. In particular, we examine whether cloud services, fog nodes, end devices, and their specific properties can be modelled. This includes the mobility of end devices and fog nodes, which may change their physical location. Moreover, different threats to data protection may arise from network configurations. Thus, the possibility of representing relevant network properties is also examined.

Second, we look into the specifics of data protection modelling in the scope of fog computing. On the one hand, both languages are designed to model information security and not data protection. On the other hand, new data protection aspects arise with fog computing. To this end, it is examined whether data storage, transmission and processing as well as the sensitivity of data can be represented. Since specific roles are distinguished by the GDPR and different actors can have different levels of trustworthiness, it is also interesting to see if these aspects can be captured in UMLsec and SysML-Sec. Lastly, we investigate whether data protection mechanisms such as encryption or access control can be modelled.

To answer the research question, we create models using UMLsec deployment and class diagrams as well as SysML-Sec requirement and parametric diagrams. These diagram types complement each other to represent multiple perspectives of data protection aspects. With each of the four diagram types we model all three use cases described in Sec. IV. The modelling is based on textual and graphical descriptions from [14] as well as on documentation and discussions within FogProtect. Data protection aspects were identified in [14] but only described in natural language.

Diagram extracts from various use cases are used to illustrate data protection modelling in Sec. V. Full diagrams can be found online[3]. In the modelling process, we pay special attention to modelling decisions based on limitations of the modelling languages. The modelling process and the diagrams are examined in Sec. VI and possible extensions of both languages are proposed.

IV. Use Cases

The EU research project FogProtect [12] aims at delivering new and advanced architectures, technologies, and methodologies to ensure data protection, from cloud centres through fog nodes to end devices[4]. The use cases modelled in this paper stem from FogProtect and are the following [15].

Smart City. Cameras monitor selected places in a city to detect incidents like car accidents. Incidents can also be reported by citizens via a smartphone app. The cameras are connected to fog nodes that analyse the video stream, anonymize it, and store the original video files for a certain period of time. In a cloud-based application the video stream is further processed beyond the computing capabilities of the fog nodes, before it is made available to a monitoring platform operated on a third-party cloud. Via the monitoring platform, it is possible to watch the anonymised video stream. The raw video material can be requested by an authorised person (e.g. law enforcement officer) from the respective fog node. Data protection risks could arise especially during the transmission and storage of the raw video material and during the video analysis by external parties. Actions must also be taken to ensure that the fog nodes are protected from physical attacks.

Smart Manufacturing. A mobile production area is deployed in a physical container called "Factory in a Box" (FiaB). Both end devices (including cameras, sensors, robots) and fog nodes are operated in the FiaB. Humans can also work in the FiaB. Furthermore, there is a connection to a cloud (called FiaB Cloud), which manages one or more FiaBs so that they can also be used remotely. Fog nodes process and store data produced by the end devices or received from the cloud. The FiaB Cloud is connected to a cloud operated by a third party, where services such as order management are performed. Data access via a dashboard is possible both from within the container and via the FiaB Cloud. Data protection is particularly at risk when transferring and storing personal data such as authentication data and video recordings from

[3]See https://drive.google.com/drive/folders/1d2it_HVWHR4p12_buPb72U_IadHrxZl_ for full diagrams.

[4]See https://fogprotect.eu/ for further information.

TABLE I
DATA PROTECTION ASPECTS AND THE CORRESPONDING FOGPROTECT USE CASE IN WHICH THEY OCCUR (MARKED WITH X).

Data protection aspect	Smart City	Smart Manufacturing	Smart Media
Data theft			
Personal data theft	x	x	x
Theft of authentication data		x	
Theft of intellectual property		x	
Theft of customer relevant data		x	x
Data manipulation			
Manipulation via a dashboard		x	
Corruption of data		x	
Unauthorised data access			
Access via a dashboard		x	
Too extensive data access			x
Untrusted provider			
Untrusted IaaS provider			x
Untrusted PaaS provider	x		
Untrusted SaaS provider		x	

the container. There is also a risk of data being read or manipulated by unauthorised persons via the dashboards.

Smart Media. End devices (a camera in a portable video booth or a smartphone) are used to record video interviews, which are then processed and edited into a film. Interview questions are provided via a service operated in the cloud. The video material is processed at both cloud and fog level. A fog node is operated within the video booth, which first processes videos and then forwards both videos and metadata to the databases in the cloud. On the one hand, metadata might include personal information like name, date of birth and contact information of the interviewee. On the other hand, metadata can include emotions of the interviewee, which are extracted from the video with the help of AI services. Video editing programs are run on a second cloud. The video database can be accessed by human video editors and videos can be selected using search filters based on metadata. Data protection risks could arise in the transmission and storage of metadata. In addition, it is important to ensure that third-party providers do not gain access to data and that insiders can only access data for which they are authorised. For example, the person responsible for the interview questions (chatterbox manager) and the video editor may only access selected data.

V. MODELLING DATA PROTECTION

Table I shows that the use cases cover several different types of data protection aspects that were defined in [14]. In the following subsections, we show extracts of the UMLsec and SysML-Sec diagrams that we created for the use cases to demonstrate how these data protection aspects can be modelled. The diagram extracts were chosen so that, if possible, they depict several data protection aspects. Data protection aspects include both data protection requirements and data protection threats. This is because a threat can also be captured as a requirement to prevent the given threat, and vice versa.

A. UMLsec

Regardless of specific data protection aspects, UMLsec class diagrams must be annotated with «secure dependency» to

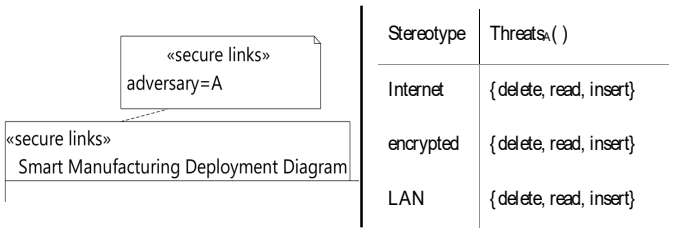

Fig. 1. Deployment diagram annotated with stereotype «secure links» (left) and adversary table for adversary "A" (right).

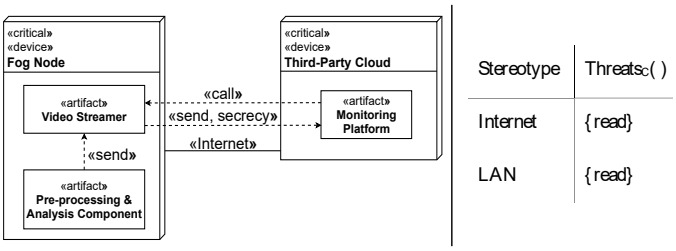

Fig. 2. Modelling data protection threat "personal data leakage" in an UMLsec deployment diagram (left) and adversary table for adversary "C" (right).

model data protection aspects. For the same reason, UMLsec deployment diagrams have to be annotated with «secure links». Additionally, an adversary table is defined for each annotated deployment diagram to define the potential threat of an adversary based on the specific use case. Fig. 1 shows the «secure links» annotation and the adversary called "A" who is defined in [14] as someone who threatens Internet, VPN and LAN connections by reading, manipulating or deleting data.

In the following, we go through the categories of Table I.

Data theft: The theft of personal data could happen in all of the three use cases. For example, in the Smart City use case, raw video data could be stolen by an adversary while it is transferred via an internet connection from the *Video Streamer* software hosted on the *Fog Node* to the *Monitoring Platform* software hosted on the *Third-Party Cloud*. Also the raw video data is in danger because the *Fog Node* does not offer sufficient security while storing the data.

The requirement that personal data should be protected from read access while transferring data between the *Video Streamer* and the *Monitoring Platform*, can be modelled in a deployment diagram by annotating the dependency between the two software components with the stereotype «secrecy». The internet connection, representing the security mechanism, is modelled by annotating the link between the *Fog Node* and the *Third-Party Cloud* with the stereotype «Internet», as shown in Fig. 2. In combination with the respective adversary table (Adversary "C" in the Smart City use case), the read vulnerability of data transfer via an internet connection can be observed. The data protection threat resulting from insufficient security at the *Fog Node* cannot be modelled in the deployment diagram because none of the possible stereotypes («smart card», «POS device», «issuer node») suits a fog node.

Personal data theft can also be modelled in an UMLsec

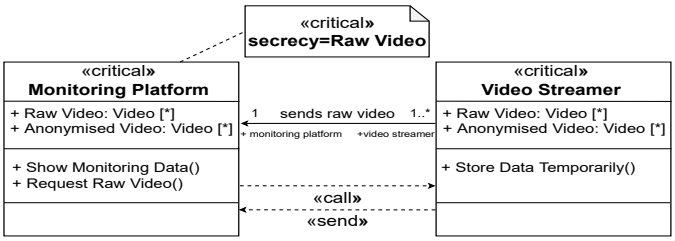

Fig. 3. Modelling the data protection threat "personal data leakage" in an UMLsec class diagram.

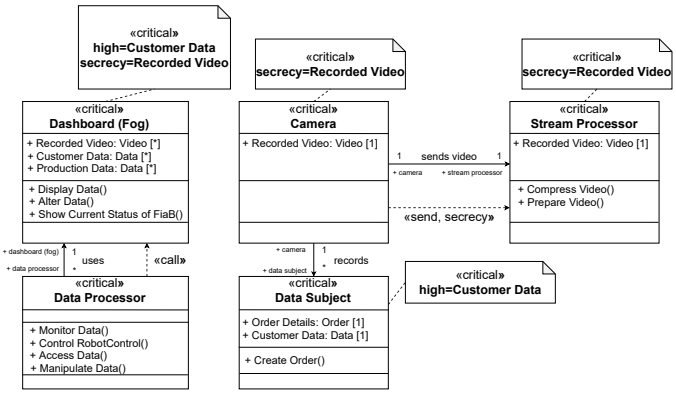

Fig. 4. Modelling actor involvement in an UMLsec class diagram.

class diagram. This has the advantage that it is possible to specify which personal data (e.g., *raw video data*) is at risk while abstracting from the technical realisation of data transfer (e.g., internet connection). Fig. 3 shows the Smart City class diagram. It can be seen that a dependency between the classes *Monitoring Platform* and *Video Streamer* exists that is annotated with the stereotype «send». Moreover, both classes are annotated with the stereotype «critical». At the *Monitoring Platform*, the property *Raw Video* is tagged with {secrecy}. This model extract in combination with the class diagram being annotated with the stereotype «secure dependency» reveals two issues. First, the class *Video Streamer* does not deliver sufficient security because the property *Raw Video* is not tagged with the same value as it is at the class *Monitoring Platform*. Second, the dependencies do not offer sufficient security because they are not annotated with «secrecy».

An exception from modelling personal data theft with «secrecy» (deployment diagram) and {secrecy} (class diagram) is made for data protection threats "theft of authentication data", "theft of intellectual property" and "theft of customer relevant data" from the Smart Manufacturing use case. For example, authentication data could be stolen when it is transferred via an unsecured connection from the end device called *Authentication Module* to the *Data Hub*, if someone gains physical access to the *Authentication Module* or if the authentication data is stored insecurely in the *Data Hub*. Similarly, authentication data could also be manipulated or deleted in this use case. Therefore, the stereotype «high» is used instead of «secrecy» in both deployment and class diagrams whenever multiple data protection threats (theft, manipulation, deletion) could occur. The corresponding adversary table can be seen in Fig. 1.

Data manipulation: To model data manipulation, two stereotypes can be used. «integrity» could be used when there is a risk that data may be changed without permission. It corresponds to the "insert" threat, while data deletion can be mapped to the "delete" threat in an adversary table. To model that potential deletion risks exist, the stereotype «high» can be used because it is the only one that corresponds to the value "delete" in the adversary table.

In the Smart Manufacturing use case, data manipulation may occur in two different ways. First, data can be manipulated by using a dashboard. The *Dashboard* allows actors to change personal data. This may be exploited by an adversary that gains unauthorised access to the *Dashboard*. The threat of unauthorised physical access could be modelled in a deployment diagram by combining the "access" threat value in an adversary table with annotating the *Dashboard* with a suitable stereotype. However, none of the stereotypes readily available in UMLsec are suitable for this purpose. In a class diagram, it is possible to model an actor as a class (see *Data Processor* and *Data Subject* in Fig. 4). However, the existing stereotypes and tags of UMLsec do not allow to model the threat of unauthorised access to the *Dashboard* in a meaningful way.

Second, data deletion could occur when data is transferred or stored with insufficient security. As already explained related to the example of the threatened authentication data in the Smart Manufacturing use case, the stereotype «high» is used to model a data deletion threat.

Unauthorised data access: This threat means that unauthorised actors can access (read) data they are not allowed to see. In a deployment diagram, the "read" threat can be modelled by using the stereotype «secrecy». However, actors and data access by an actor cannot directly be modelled in a deployment diagram. Using the "access" threat in combination with an annotated component only means that adversaries may access components they are not allowed to access. Neither actors nor data can be represented in a deployment diagram. In class diagrams, it can be modelled that actors represented by a class have a dependency, representing data access, to other classes (see Fig. 4). However, it is not possible to show that an actor has too extensive data access.

Untrusted provider: Similar to the threat of unauthorised data access, modelling the threat to data protection stemming from an untrusted provider (Infrastructure, Platform or Software as a Service provider) is not possible in a meaningful way. In a class diagram, trustworthiness could be represented by adding a property called "trustworthy" to a class representing an actor. However, this does not have the same impact as a stereotype (e.g., for verifying whether security requirements are met). Therefore, we decided against it.

B. SysML-Sec

The requirement and parametric diagrams modelled in SysML-Sec are structured according to the same scheme in

all three use cases. Therefore, modelling of data protection requirements and modelling of potential attacks on personal data are each described by using an example diagram extract.

Modelling data protection requirements: As can be seen in Fig. 5, the requirement diagram is structured into multiple levels. Every requirement is annotated with the stereotype «SecurityRequirement». The requirement on top of each requirement diagram is called *PreventDataProtectionViolation*. This covers the high-level requirement that breaches of the GDPR should be prevented. The requirement is assigned the type "Privacy", and the risk that the requirement will be violated is classified as "High". Both the extent of damage and the probability of occurrence are included in the risk assessment. It is possible to assign an attack from the parametric diagram to each security requirement. Since this top-level requirement is very abstract, no attack is assigned to it.

At the next level are security requirements that relate to potential threats to data protection. Fig. 5 shows that the requirements on this level are connected to the top-level requirement by using a "containment relationship" (plus sign in a circle at the end of a line). By using the "containment relationship", the top-level requirement is broken down into sub-requirements. Thus, they form a hierarchy in which the sub-requirements must be fulfilled in order for the top-level requirement to be fulfilled.

Fig. 5 shows the sub-requirement *PreventAccessThroughUntrustedParties*. This security requirement refers to the risk that untrusted actors, such as IaaS providers, could access data. The requirement defines that such access should be prevented. "Controlled access (authorisation)" was selected as the "kind", since access to data is controllable via authorisation. The risk is classified as "medium", because the extent of damage is assessed as high and the probability of occurrence as low.

The next hierarchy level is associated with the «deriveReqt» relationship. The difference to a "containment relationship" is that in a «deriveReqt» relationship the main requirement describes the WHAT and the sub-requirement describes the HOW. In a "containment relationship", specification takes place at the same level of abstraction. The OMG describes "containment relationships" as a tool to decompose complex requirements into simpler, single requirements, while a «deriveReqt» relationship includes a hierarchy level change[5].

The sub-requirement shown in Fig. 5 is titled *UseOfSecureStorage*. The requirement is fulfilled if all components in the Smart Media use case that store personal data are protected. Accordingly, *UseOfSecureStorage* is broken down by "containment relationships" into requirements that relate to the individual data-storing components. Both main and sub-requirements are assigned the type "Confidentiality", as they relate to the risk of confidentiality violation. The risk of *UseOfSecureStorage* being violated is classified as "High". The risk is derived from the highest risk posed by the breach of one of the sub-requirements. On the lowest level of the

[5]See https://www.omg.org/spec/SysML/1.6/ for further information.

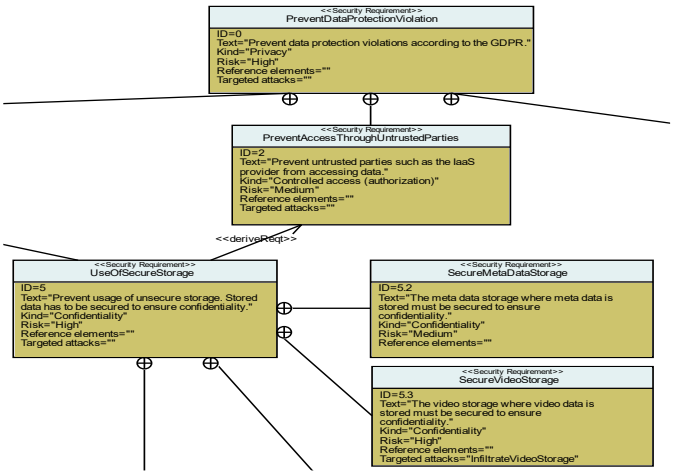

Fig. 5. Excerpt from the Smart Media requirement diagram.

requirement diagram, it is possible to assign a respective attack from the parametric diagram to the requirements.

Modelling potential attacks on personal data: An attack tree was created for each identified attack on personal data by using SysML-Sec parametric diagrams. Fig. 6 shows an excerpt of the Smart Media parametric diagram. The root of the attack tree is called *AccessMetaData* and is annotated with the stereotype «root attack». This root attack is divided into several attacks by the logical operator «OR». If one of the sub-sequences is executed successfully, the root attack is also executed successfully. The temporal operator «SEQUENCE» states that the attacks are carried out in a temporal sequence from left to right. The attacks on the right side of the sequence can only be carried out if the previous attacks were successful. Attacks are annotated with «attack».

The attack sequence starts with the attack *InfiltrateConnectionBetweenSmartphoneAndChatterboxService*. If the attack was successful, then the *ReadData* attack is executed. Both attacks are modelled inside of a «block» called *Smartphone*. This «block» is inside of another «block» called *ChatterboxCloud*. Blocks represent hardware components. Nested blocks were used to model attacks that belong to multiple hardware components (e.g., infiltration of connections). The second attack sequence refers to the *ChatterboxCloud*. Therefore, the attacks are only inside of one «block».

VI. Discussion

A. Evaluation of the modelling results

To answer the research question, the modelling results are evaluated. An overview is shown in Table II.

The first part of the *RQ* can be answered as follows. Both modelling languages can be used to model fog computing systems, but limitations arise that prevent modelling special fog computing characteristics explicitly. The same applies to the second part of the *RQ*. Many data protection aspects can be modelled with the modelling languages intended for information security modelling. Nevertheless, limitations also occur in this field. The details are explained below.

TABLE II
REPRESENTABLE FOG COMPUTING DATA PROTECTION ASPECTS IN UMLSEC AND SYSML-SEC

Representable aspect	UMLsec deployment and class diagrams	SysML-Sec requirement and parametric diagrams
To what extent can UMLsec and SysML-Sec serve as a basis for modelling fog computing systems?		
Cloud services, fog nodes, end devices and their specific properties	• Deployment diagrams: «device» to model infrastructure, «artifact» to model software components • Class diagrams: Infrastructure and software components modelled as classes	• Requirement diagrams: Natural language text but no explicit modelling construct • Parametric diagrams: Blocks representing the target of an attack
Location of end devices and fog nodes	• Location as an attribute inside of class diagrams • Dynamic change of the location is not representable	• Natural language text but no explicit modelling construct in requirement diagrams
Network properties	• Limited to stereotypes describing the connection type in deployment diagrams	• Natural language text but no explicit modelling construct in requirement and parametric diagrams
To what extent are UMLsec and SysML-Sec suitable for modelling fog computing related data protection aspects?		
Secure data storage, transmission and processing	• Deployment diagrams: Secure data transmission • Class diagrams: Secure storage (properties), processing (operations), transmission (dependencies)	• Natural language text but no explicit modelling construct in requirement and parametric diagrams
Data sensitivity	• Combination of stereotype «secrecy» and the tags {secrecy}, {integrity}, {high} in class diagrams	• "Kind" attribute of a security requirement in requirement diagrams
Actors and their respective role in relation to personal data	• Indirectly modelled in class diagrams	• Natural language text but no explicit modelling construct in requirement diagrams
Trustworthiness of actors	• Not possible in a meaningful way	• Natural language text but no explicit modelling construct in requirement diagrams
Presence of data protection mechanisms	• Stereotypes and tags in deployment and class diagrams can be used • Only the level of protection and not the data protection mechanism itself can be meaningfully represented	• Requirement diagrams: Natural language text but no explicit modelling construct • Parametric diagrams: «countermeasure»

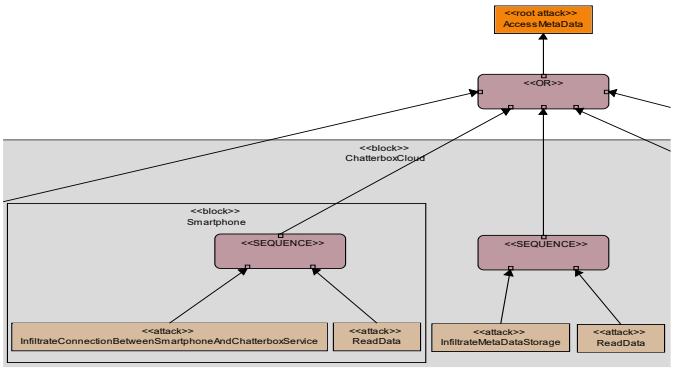

Fig. 6. Excerpt from the Smart Media parametric diagram.

UMLsec: Using UMLsec, it is possible to represent cloud services, fog nodes and end devices in deployment and class diagrams, although the modelling language is not intended specifically for use in fog computing. For example, in deployment diagrams the infrastructure components mentioned in the use cases can be modelled as a component annotated with the stereotype «device». Software components that are operated on the respective infrastructure components can be annotated with the stereotype «artifact». In a class diagram, both infrastructure and software components can be modelled as classes. The properties of the respective components can be modelled in class diagrams through relations and dependencies to other components as well as through properties and operations.

UMLsec class diagrams can model the location of devices as a property, but the dynamic change of the location cannot be meaningfully represented. For example, in the Smart Media use case, the *HDD Storage* is physically moved from the *Computer* to the *Chatterbox Cloud* to copy the data via a wired connection. Logically, the *HDD Storage* can only be connected to one of the two devices at the same time. This cannot be represented in UMLsec deployment and class diagrams.

The network properties modelled in UMLsec deployment diagrams are limited to stereotypes specifying connection types between the infrastructure components. Other network properties such as bandwidth and latency are not part of the use cases and are therefore not modelled.

UMLsec deployment diagrams show whether connections between infrastructure components are protected from potential adversaries. Adversaries and different connection types can be modelled by using the adversary table and by annotating connection links as well as dependencies.

In UMLsec class diagrams, storage (properties), processing (operations) and exchange of data (dependencies) can be modelled. Specifically the transmission of data cannot be represented though. In order to mark personal data, classes can be annotated with the stereotype «critical» and properties can be assigned an existing protection with the tags {secrecy}, {integrity} and {high}. It can then be checked whether the required protection is present during the transmission and storage of personal data.

Actors and their roles in relation to the data cannot be modelled in UMLsec deployment diagrams and only indirectly be modelled in UMLsec class diagrams. Also, trustworthiness of actors cannot be meaningfully represented.

To model existing data protection mechanisms in UMLsec, stereotypes and tags could be used, but these UMLsec concepts only describe the level of protection and not the type of data protection mechanism. Therefore, specific data protection mechanisms cannot be meaningfully represented.

TABLE III
OVERVIEW OF THE ENCOUNTERED LIMITATIONS

UMLsec
• Missing stereotypes to model wireless data transfer
• Missing stereotypes to model the threat of unauthorised accessing fog nodes or end devices by an adversary
• No option to model threats between components on the same hardware
• Problems at modelling rights and obligations of data-related roles in relation to data because classes cannot be annotated meaningfully
• Missing stereotype to model trustworthiness of actors
• No meaningful possibility to model actors with multiple roles
• No stereotype to model an isolated data deletion threat
SysML-Sec
• No "kind" attribute value for modelling the trustworthiness of actors
• No stereotypes to assign blocks to cloud, fog or end device layer
• Problems at assigning attacks to multiple components

SysML-Sec: Due to the freely usable text fields "title" and "description", it is possible to represent fog computing concepts in the requirement diagram by modelling requirements annotated with «SecurityRequirement». Attacks and blocks can also be freely named in the parametric diagram. This makes it possible to distinguish between end devices, fog nodes and cloud services. However, new stereotypes with which, for example, the blocks could be annotated, would allow a more formal modelling beyond natural language.

The requirement diagram can be used to model data protection requirements that relate to both personal data and the existence of data protection mechanisms. On the one hand, the title and textual description of security requirements define the security requirement. On the other hand, the attributes "kind" and "risk" can be used to specify the security requirements. Only the trustworthiness of actors cannot be directly represented in a meaningful way because actors and roles are not represented and there is no "kind" value for trustworthiness.

Existing data protection mechanisms can be modelled as countermeasures («countermeasure») in attack trees in parametric diagrams. Security requirements from requirement diagrams that are related to protection mechanisms can be linked to potential attacks by using the attribute "Targeted attacks".

Secure storage, processing and transmission of personal data can be modelled as security requirements. The roles of actors in relation to data and the trustworthiness of actors can be shown indirectly through security requirements. SysML-Sec does not allow for the explicit modelling of data-related roles of actors and their trustworthiness. The same applies to the explicit modelling of data and the transmission of certain data.

B. Encountered limitations

To deal with the both, already indicated and further limitations, modelling decisions had to be made. An overview of the encountered limitations can be found in Table III.

UMLsec: In the future the connection between the *CCTV Camera* and the *Fog Node* in the Smart City use case should be realised wireless. An UMLsec deployment diagram does not support stereotypes representing wireless connections. Therefore, wireless data transfer is not explicitly modelled.

UMLsec deployment diagrams offer the possibility to annotate nodes with the stereotypes «LAN», «smart card», «POS device» and «issuer node». In combination with the "access" value inside of the threat list of an adversary table, the threat of physical access by an adversary could be modelled. All available stereotypes are very specific and do not fit an end device or fog node from the FogProtect use cases. Therefore, physical access could not be modelled.

Data protection threats arising from the communication of software components within one infrastructure component can only be modelled to a limited extent. For example, Fig. 2 shows two software components within the «device» *Fog Node* that exchange personal data with each other. To model data protection threats in the UMLsec deployment diagram the dependency and the connection type between the components as well as the adversary table of the diagram are needed. Thus, it is not possible to model data protection threats that may occur between two components hosted on the same hardware, when they exchange personal data. This threat only becomes apparent when examining the Smart City class diagram.

To model the affiliation of data to a role in a UMLsec class diagram (e.g. *Recorded Video* and *Data Subject* in the Smart Manufacturing use case), data could be modelled as classes. However, classes can only be annotated with the stereotype «critical». The associated tags {secrecy}, {integrity} and {high} can only be related to properties, operations and signals [10]. Because in UMLsec the needed stereotypes and tags cannot be applied to classes it is not useful to model data as classes. Essential UMLsec concepts would thus be lost in the model. For this reason, data in the UMLsec class diagrams were modelled as public properties (example see Fig. 3). This leads to the fact that rights and obligations of data-related roles in relation to data are difficult to recognise. To identify the relation between a data subject and their personal data, information from relation names and property names has to be combined. It might be necessary to trace relationships across several classes (e.g. relationships starting from the *Data Subject* class to the *Stream Processor* in Fig. 4).

Since the class diagrams modelled in UMLsec were not modelled at instance level but at type level, properties such as the trustworthiness of a person or a company are not represented. One way to model trustworthiness is to add a boolean property *trustworthy* to the classes *Data Processor*, *Data Controller*, *IaaS / SaaS / PaaS Provider*. However, the combination of stereotypes, tags and the property *trustworthy* cannot be used in a UMLsec class diagram to make a statement about the protection of personal data. It therefore makes no sense to include this as a property. The danger of possible access to personal data by providers that are classified as untrustworthy is not recognisable in the UMLsec diagrams.

Another limitation that arises from the representation of roles as classes occurs when attempting to model actors who can take on multiple roles at once. In the Smart Manufacturing use case, the workers within the FiaB container are both *Data Subject*, as they are filmed, and *Data Processor*, as they operate the *Dashboard (Fog)*. This dual role is not visible in

the created models (see Fig. 4). With the help of a newly introduced class that inherits from *Data Subject* and *Data Processor*, it would be possible to represent the dual role. However, it does not make sense to introduce such a class at the type level, as it must be assumed that the role of the worker changes situationally at the instance level.

Lastly, to model a data deletion threat the stereotype «high» or the tag {high} can be used. However, this annotation refers to a threat including potential readability, manipulation, and deletion of the data at the same time. Therefore, it is not clear that only the threat of data deletion exists.

SysML-Sec: Since requirement and parametric diagrams allow to freely select requirement and attack titles most data protection aspects could be modelled. However, by using natural language, the advantages of a formal modelling are lost. Additional stereotypes, attributes and values for the existing attributes could improve the modelling with SysML-Sec.

The possible values that can be assigned to the "kind" attribute cannot describe the trustworthiness of actors. Therefore, the data protection risks that arise from actors classified as untrustworthy are not modelled. Trustworthiness is only implicitly modelled by a security requirement of the kind "Controlled Access (authorisation)".

Attacks that represent the infiltration of a connection between two components can only be assigned to blocks with limitations. To model this blocks have to overlap each other to assign an attack to multiple components.

C. Possible language extensions

Further development of UMLsec and SysML-Sec would facilitate their use in the areas of fog computing and data protection. The limitations could be addressed by introducing new stereotypes, tags and attribute values.

Connection types such as «WLAN», «Bluetooth», «4G» and «5G» could supplement the existing connection types in the UMLsec deployment diagram to model wireless data transmission. The extension of the stereotypes with which one can annotate a «device» would also make sense, since UMLsec only includes stereotypes specific to one subject area. Both «issuer node» and «POS device» refer to components from the field of payments. The possibility to annotate an infrastructure component as vulnerable to an attack is not provided. Such a stereotype could be called «point of attack». Another idea is to introduce special stereotypes that make it possible to distinguish between end devices, fog nodes, and cloud.

In the UMLsec class diagram, further stereotypes could be introduced to annotate classes. Among others, a stereotype called «trustworthy» could be used to represent trustworthy classes as long as this is possible on type level. With a stereotype «authorised», dependencies could be annotated to model authorised data access in combination with the stereotype «call». By enabling the annotation of classes through the stereotypes «secrecy», «integrity» and «high», data could be meaningfully represented as classes. Furthermore, IT security protection goals such as "reliability" and "authenticity" could extend the stereotypes and tags.

Security requirements in SysML-Sec requirement diagrams could be extended by further attributes. For example, it could be possible to explicitly model the trustworthiness of actors by using a new attribute "trustworthiness". Alternatively, it would also be possible to extend the values of the attribute "kind" by adding "trustworthiness" as a new value. In parametric diagrams, new stereotypes like «cloud», «fog node», «end device» could be used to specify blocks.

D. Risks to validity

Both the modelling of the use cases and the discussion of the results are exposed to validity risks.

Internal validity risks that exist in this work are interpretation errors of the modelling basis and of the modelling results as well as the risk of having applied the modelling languages incorrectly. In order to reduce internal validity risks, ambiguities were discussed in talks with the FogProtect partners. Furthermore, the FogProtect partners were asked to evaluate the diagrams and identify errors in the content. Several precautions were taken to avoid errors in the modelling: On the one hand, the modelling results of the different use cases were compared with each other. On the other hand, primary literature documenting the use of the modelling languages was employed [9]–[11], [16], [17]. By using special modelling tools that are recommended for modelling in UMLsec or SysML-Sec, syntactic errors in the models are excluded. These modelling tools are the UMLsec4UML2 profile [16] that is used within Eclipse Papyrus[6] for modelling UMLsec diagrams and TTool[7] for modelling SysML-Sec diagrams.

There is a risk to external validity that the discussion results are not generalisable because few use cases cannot cover all aspects from the fields of fog computing and data protection as well as all fog computing related data protection aspects. To mitigate this risk, multiple use cases from different fog computing domains were used instead of just one use case. Finally, it should be emphasised that it is not possible to fully evaluate all possible scenarios and associated data protection aspects with UMLsec and SysML-Sec.

VII. Related Work

There are some approaches in the literature for modelling IoT or cloud specific information security aspects. For instance, IoTsec can be used to model security issues in IoT systems by combining concepts of UML, UMLsec and SysML [18]. Chambwe introduced ThingMLsec, an extension of UMLsec that adds domain-specific concepts related to IoT applications [19]. Ficco et al. introduced new stereotypes to UMLsec to model a secure deployment of cloud applications [20]. But, the approaches do not consider modelling IoT and cloud elements as part of fog computing systems. Also, aspects of data protection beyond information security are not covered.

Concerning the modelling of data protection concerns, Ramadan proposed a framework that includes SecBPMN2, UMLsec and UMLfair to assure data protection by design [21].

[6]See https://www.eclipse.org/papyrus/ for further information.

[7]See https://ttool.telecom-paris.fr/ for further information.

There, the UML rabac profile, that extends UMLsec, is used to model role- and attribute based access control [22]. Access control can also be modelled by using PrivUML, a modelling language to model privacy protection [23]. A related UML profile to model privacy-aware applications also exists [24]. Moreover, an UMLsec extension to model legal regulations based on information security law and ISO/IEC 27001:2005 also allows to model data protection [25]. None of these approaches allow to model cloud or fog computing related data protection aspects, such as limited personal data protection at fog nodes. But, they could serve as an additional basis to complement the proposed extensions.

Some approaches allow to model data protection in cloud systems. Palm et al. proposed a risk pattern based approach to model data protection vulnerabilities [26]. Shei et al. defined a language capable of modelling cloud computing concepts as well as information security requirements that also relate to data protection [27]. These approaches do not cover characteristics of fog computing, nor are they as extensive as UMLsec or SysML-Sec.

VIII. Conclusion

In this work we have examined to what extent UMLsec deployment and class diagrams as well as SysML-Sec requirement and parametric diagrams can be used to model data protection aspects related to fog computing.

It has become clear that both modelling languages are a good basis for modelling data protection aspects related to fog computing. With the help of this type of modelling, it is possible to evaluate, among other things, whether data protection is being sufficiently observed at the design time. However, several limitations (e.g. missing stereotypes to model specific data protection aspects like trustworthiness of actors; no explicit modelling of personal data) were identified. In order to mitigate these limitations we proposed extensions to both UMLsec and SysML-Sec.

We assume that the results can also be applied to related domains such as the Internet of Things, cloud computing, edge computing. Further research could verify this thesis. This work can serve as a basis for finding further extensions to UMLsec and SysML-Sec and also for implementing the proposed extensions. Moreover, an examination of whether other UMLsec and SysML-Sec diagram types are feasible for modelling data protection aspects related to fog computing could extend our work. It should also be considered that the use of additional or other modelling languages (and the combinations of languages) may address the limitations.

References

[1] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, pp. 289–330, 2019.

[2] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, Mobidata '15, pp. 37–42, 2015.

[3] C. Avasalcai, I. Murturi, and S. Dustdar, "Edge and fog: A survey, use cases, and future challenges," in *Fog Computing: Theory and Practice*, pp. 43–65, 2020.

[4] Y. Guan, J. Shao, G. Wei, and M. Xie, "Data Security and Privacy in Fog Computing," *IEEE Network*, vol. 32, no. 5, pp. 106–111, 2018.

[5] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *International Conference on Wireless Algorithms, Systems, and Applications*, pp. 685–695, Springer, 2015.

[6] General Data Protection Regulation, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC," *Official Journal of the European Union*, p. L119, 2016.

[7] Z. Á. Mann, F. Kunz, J. Laufer, J. Bellendorf, A. Metzger, and K. Pohl, "RADAR: Data protection in cloud-based computer systems at run time," *IEEE Access*, vol. 9, pp. 70816–70842, 2021.

[8] T. Wettig and Z. Á. Mann, "Simulation-based analysis of threats to location privacy in fog computing," in *IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pp. 736–741, 2021.

[9] J. Jürjens, "UMLsec: Extending UML for secure systems development," in *Proceedings of the 5th International Conference on the Unified Modeling Language*, vol. 2460, pp. 412–425, 2002.

[10] J. Jürjens, *Secure Systems Development with UML*. Springer, 2005.

[11] L. Apvrille and Y. Roudier, "SysML-sec: A sysML environment for the design and development of secure embedded systems," in *APCOSEC, Asia-Pacific Council on Systems Engineering*, 2013.

[12] D. Ayed, E. Jaho, C. Lachner, Z. Á. Mann, R. Seidl, and M. Surridge, "FogProtect: Protecting Sensitive Data in the Computing Continuum," in *Advances in Service-Oriented and Cloud Computing*, pp. 179–184, 2021.

[13] International Organization for Standardization, "Information technology – Security techniques – Information security management systems – Overview and vocabulary(2700:2018)," standard, Feb. 2018.

[14] FogProtect, "D2.1: State of the Art Analysis and Initial Validation Plan." https://fogprotect.eu/deliverables/d2-1-state-of-the-art-analysis-and-initial-validation-plan/, 2020.

[15] K. Baert, J. Garcia, J. Kuhr, E. Salant, R. Seidl, and R. Vitorino, "End-To-End Data Protection Through the Computing Continuum in Smart Environments," *2021 Joint EuCNC & 6G Summit*, 2021.

[16] H. Schmidt and J. Jürjens, "UMLsec4UML2 - Adopting UMLsec to Support UML2," Department of Computer Science, Technische Universität Dortmund, 2011.

[17] L. Aprville, "SysML-Sec Tutorial," Télécom ParisTech, 2020.

[18] D. A. Robles-Ramirez, P. J. Escamilla-Ambrosio, and T. Tryfonas, "IoT-sec: UML Extension for Internet of Things Systems Security Modelling," in *2017 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, pp. 151–156, 2017.

[19] K. K. Chambwe, "Model-based Secure Software Engineering using UMLsec applied to Assisted Living and Home Care," master thesis, Institutt for informatikk, University of Oslo, 2018.

[20] M. Ficco, F. Palmieri, and A. Castiglione, "Modeling Security Requirements for Cloud-based System Development," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 8, pp. 2107–2124, 2015.

[21] Q. Ramadan, "Data Protection Assurance by Design: Support for Conflict Detection, Requirements Traceability and Fairness Analysis," phd thesis, Institut für Softwaretechnik, Universität Koblenz-Landau, 2020.

[22] A. S. Ahmadian, D. Strüber, V. Riediger, and J. Jürjens, "Model-based Privacy Analysis in Industrial Ecosystems," in *European Conference on Modelling Foundations and Applications*, pp. 215–231, Springer, 2017.

[23] J. El Mokhtari, A. Abou El Kalam, S. Benhadou, and H. Medroumi, "PrivUML: A privacy metamodel," *Procedia Computer Science*, vol. 151, pp. 53–60, 2019.

[24] T. Basso, L. Montecchi, R. Moraes, M. Jino, and A. Bondavalli, "Towards a UML profile for privacy-aware applications," in *IEEE International Conference on Computer and Information Technology*, pp. 371–378, 2015.

[25] S. Islam and J. Jürjens, "Incorporating security requirements from legal regulations into UMLsec model," *Modelling Security Workshop (MODSEC08)*, vol. 8, 2008.

[26] A. Palm, Z. Á. Mann, and A. Metzger, "Modeling data protection vulnerabilities of cloud systems using risk patterns," in *System Analysis and Modeling*, pp. 1–19, 2018.

[27] S. Shei, C. Kalloniatis, H. Mouratidis, and A. Delaney, "Modelling secure cloud computing systems from a security requirements perspective," in *Trust, Privacy and Security in Digital Business*, pp. 48–62, 2016.