UNIVERSITY OF TARTU
Institute of Computer Science
Software Engineering Curriculum

**Eduard Sing**

# A Meta-Model Driven Method for Establishing Business Process Compliance to GDPR

**Master's Thesis (30 ECTS)**

Supervisor(s): Raimundas Matulevičius
Jake Tom

Tartu 2018

# A Meta-Model Driven Method for Establishing Business Process Compliance to GDPR

**Abstract:**

In the April 2016, the European Parliament and Council approved the new personal data protection regulation - GDPR (General Data Protection Regulation), which will take effect at the end of the May 2018 in all Member States of European Union (EU). The GDPR is addressing common problems of the protection and the usage of the personal data of EU citizens. According to the new regulation, all organizations that use personal data of EU citizens in their day-to-day activities - have to re-evaluate their business processes and information systems to comply with the new rules and constraints. The punishment for misuse of personal data can be very costly to the company - up to 20 million euros or 4% of the annual global turnover in fines. Nevertheless, there is no technical guidance or clear approach that would help to evaluate business processes of an information system to comply with GDPR. This thesis will address mentioned issue by researching the GDPR legislation text and proposing an actual methodology for analysing business processes of information systems and aligning them with the GDPR. The proposed methodology will also help to map the flow of the personal data between different parties and highlight the problematic places in the business processes suggesting measures to reduce the misuse of personal data. This approach could be used as a reference point for developing the automated tool for analysing the processes of an information system to comply with GDPR.

**Keywords:**

GDPR, General Data Protection Regulation, Personal Data, compliance, Information System, business process

**CERCS:**

**T120 -** Systems engineering, computer technology

## Metamudelile toetuv meetod äriprotsessi vastavusse viimiseks Euroopa Liidu isikuandmete kaitse üldmääruse nõuetega

**Lühikokkuvõte:**

2016. aasta aprillis kiitis Euroopa Parlament ja Nõukogu heaks ning võttis vastu uue isikuandmete kaitse määruse - GDPRi (Isikuandmete kaitse üldmäärus), mis jõustub 2018. aasta mai lõpus Euroopa Liidus (EL). GDPRi eesmärgiks on lahendada ELi kodanike isikuandmete kaitse ja kasutamisega seotud päevakohaseid probleeme. Uue määruse kohaselt kõik organisatsioonid, mis kasutavad ELi kodanike isikuandmeid oma igapäevases tegevuses, peavad oma infosüsteeme ja äriprotsesse ümber hindama, et need vastaksid uutele eeskirjadele ja piirangutele. Isikuandmete väärkasutus võib ettevõttele olla väga kulukas - kuni 20 miljonit eurot või 4% aastasest käibest trahvidena. Sellele vaatamata puudub tehniline juhis või selge lähenemisviis, mis aitaks hinnata infosüsteemide äriprotsesside vastavust GDPRi nõuetele. Käesolev töö käsitleb mainitud probleemi, uurides üldmääruse õigusakti teksti ja pakkudes välja infosüsteemide äriprotsesside analüüsimise metoodikat, mis aitaks viia äriprotsesse vastavusse GDPRi nõuetele. Pakutud metoodika aitab kaardistada isikuandmete liikumist erinevate osapoolte vahel ja tuua välja äriprotsessi probleemsed kohad, mis aitab vähendada isikuandmete kuritarvitamist. Pakutud metoodikat saab kasutada ka automatiseeritud tööriista väljatöötamiseks.

**Võtmesõnad:**

GDPR, Isikuandmete kaitse üldmäärus, isiklikud andmed, äriprotsess, infosüsteem

**CERCS:**

**T120 -** Süsteemitehnoloogia, arvutitehnoloogia

# Table of Contents

## Table of Figures

# Index of Tables

# 1 Introduction

Intensifying social integration and economic relations between the Member States of the European Union (EU) multiplied by rapid technological growth are significantly increasing use and exchange of the personal information of natural persons across the Union by both state authorities and private organizations. As a result of this, Member States, European Parliament and Council are facing new challenges in the protection of the personal data, free flow of the information between the Member States, export of data to third countries and standardization of use, collection, and storage of the information. To address these issues European Parliament and Council are enforcing new regulation - General Data Protection Regulation (GDPR). GDPR would ensure high-level standards of personal data protection and standardize its use across the Union for all Member States [1].

With the enforcement of Regulation, both private organizations and public authorities, who are using personal data in their information systems (IS) to pursue their day-to-day activities, are facing challenges to meet new requirements set by GDPR. In order to maintain high-level standards of personal data protection and tackle misuse of the information, organizations are required to perform audits of the information systems and their business processes to ensure that compliance towards GDPR is met. If a party fails to comply with new rules it can be fined up to 20 million euros or 4% of the annual global turnover [1]. Regulation is not suggesting technical guidance or any methodology of how actual information system or its business processes are expected to process the personal data. This raises next question - **How business processes of the information system should be checked for GDPR compliance?**

This work will address raised issue by researching legislation text and defining meta-model of GDPR. This meta-model will serve as the main input in the analysis of the business processes of the information system, while second input will be as-is compliancy model of the business process itself. This approach will help to highlight diversity in two models, which would help to define measures and steps to align business process with the GDPR. To illustrate and validate the suggested approach, actual business processes from information system being developed will be used.

This work is divided into six chapters as follows. The first chapter gives an overall overview of the work and defines research questions of the thesis. The second chapter gives an overview of the GDPR and similar data protection acts, discusses related work, provides the basis for the definition of GDPR meta-model and proposes it. The third chapter introduces the method of checking compliance in information systems by explaining how business processes of information systems can be mapped to proposed meta-model and later validated. In the fourth chapter, we will discuss the possibility of GDPR analysis automation and give an example of how it can be done by developing prototype tool, which will be based on the work of previous chapters. The fifth chapter will discuss the validation of the proposed method. The last, sixth chapter, concludes work, discusses limitations, learned lessons and talks about possible related future work.

## 1.1 Research Questions

The main research question (MRQ) is:

**MRQ – How business processes of the information system should be checked for GDPR compliance?** The main research question embodies several sub-questions (SUBQ):

**SUBQ1 – What is the GDPR and how to formalise it?** We will research the GDPR legislation text, find the key points and aspects that are applicable to information system

business processes. In addition to this, we will formalise regulation using UML[1] class diagram notation to propose GDPR meta-model, which will support our method to check compliance of business processes in information systems.

**SUBQ2 – What is the method to check compliance of the business process to GDPR?** Based on our findings in the first sub-question, we will propose a methodology that will allow checking the IS business processes compliancy towards GDPR.

**SUBQ3 – What is the proof of concept to support process compliance to GDPR?** We will show how methodology, discussed in second sub-question, could be used with real-world scenario business processes, moreover, we will develop the tool, which would support our findings and automate the application of the methodology.

**SUBQ4 – What are the performance and usability results of proposed methodology (meta-model-based) compared to another method (expert opinion-based)?** We will conduct a cross-validation of the method with other real word scenario business processes. Analyse results of the validation and share our findings of how discussed methodology performed and could be upgraded.

## 1.2 Summary

In this chapter, we defined the scope of the work and introduced main research question. The main research question is divided into several sub-questions that will help to follow the thesis and find the answer to the main research question.
The new data protection regulation - GDPR is coming out in the May 2018 in the EU. The organizations that use personal data of EU citizens in their information systems should audit these systems and ensure that last ones comply with the regulation. The main goal of the thesis is to suggest approaches that would allow comparing meta-model of the GDPR and business process as-is models to analyse the GDPR compliance of information system.

[1] http://www.uml.org/

# 2 Background

This chapter will discuss and give an answer to our first sub-question SUBQ1 of our main research question (MRQ). Namely, we will choose a method to formalise GDPR, by introducing General Data Protection Regulation act, talking about similar legislatures that are implemented around the world and giving an overview of the related works. The main contribution to this part would be the analysis of the legislative text of GDPR, which will help to define the meta-model of regulation, which will be used as a basis for a proposed method to check business process compliance.

## 2.1 Related Work

Several governments have introduced analogous regulations to protect personal data of their citizens, such as US Federal Trade Commission Act (FTCA), UK Data Protection Act 1998 (DPA) and 2013 Personal Data Protection Act (PDPA) (Malaysia) [2] and directive to be replaced by GDPR – Directive 95/46/EC [3].
To support our work, we conducted a literature review of works that mention at least one data protection regulation from the list above or assesses the compliance of business processes towards regulatory requirements. The results of the literature review can be divided into four categories:

### 2.1.1 Compliance with Regulations through Privacy Standards

Studies from this block are sharing a common idea of compliance of information systems to privacy standards through adoption of the principles of privacy-by-design approach.
Gan *et al*. [2] made an exploratory study where authors are discussing the adoptions of PETs (Privacy Enhancing Technologies) related to enforcement of similar to GDRP data protection regulation in Malaysia - PDPA. Gan *et al*. argue that data protection regulations and the consequence of the enforcement as the adoption of the PETs are affecting not only the organizations but also the business and working processes of organization employees, influencing the performance of business processes differently. The study shows that one possibility of compliance of business processes to similar regulations could be the adoption of PETs, however, authors are focusing mainly on the impacts of the adoptions and possible solutions for compliances, whereas the analysis of compliance of business processes received very little attention.
Robol *et al.* [4] proposed a modelling framework to support the design of GDPR compliant information systems using Socio-Technical Security method (STS). Authors believe that complex information systems can be represented as socio-technical systems, where several actors, including the IS itself, are dependent on each other to achieve common objectives, therefore the modelling of such systems can be done using STS-ml formal language. However, authors state that STS-ml lacks elements to represent several GDPR aspects, hence the language should be extended. With handful extensions, authors show how proposed framework can be used in modelling socio-technical systems to be compliant to GDPR. Nonetheless, to ensure GDPR compliancy proposed method should be used in design phases of the IS development process. And similarly, to first study the analysis of the already developed systems receives very little attention.

### 2.1.2 Business Process Compliance

Alaküla and Matulevicius [5] propose a four-step method to achieve business process compliance with standard requirements using security risk-oriented patterns while exploring insurance company business processes as a case study. The methodology proposes the instantiation of ISO27001:2013 standard as a first step, then security risk-oriented patterns

are applied to observed business processes. In last two steps, the validity of the extended business process and compliance with the standard are being checked. In this work, we propose a similar approach to check the compliance of business processes, however, we are not limiting ourselves with security risk-oriented patterns as a method to describe regulatory requirements defined in GDPR, moreover, we are instantiating a different regulatory standard (GDPR) in the first step.

### 2.1.3 Studies that Propose Frameworks to Integrate Regulatory Requirements in Information Systems

Diamantopoulou *et al*. [6] are proposing the meta-model of GDPR entities related to data protection for composing a Privacy Level Agreements (PLA) to use in e-services with the enforcement of the GDPR**.** This study states that PLAs would support and encourage EU citizens to use e-services by enhancing citizens' trust, giving the feeling of the legal protection. However, the work focus is narrowed only by design of the meta-model for composing PLAs, which focuses mainly on the concept of the "Consent" of GDPR, while the compliance of business processes in the software systems and other important aspects of GDPR are being ignored.

Becker *et al.* [7] are proposing a meta-design approach that would help to conceptualize the regulatory requirements and integrate them into information systems during the development process. As a result, the highly abstract approach that can be implemented with several iterative development processes is born. Authors distinguish two layers of integration (design and implementation) and four different viewpoints. The concept suggests moving towards regulatory requirements compliance in two different ways: from design to implementation and from implementation to design. The second approach is similar to the proposed approach in this work (*e.g.* the information system is already implemented and has to be validated against new regulatory requirements and re-designed). Despite to similarity of concepts the proposed approach is highly abstract and could be implemented with various regulations, while this work is focused on the practical adoption of GDPR compliancy requirements in given business processes of information systems.

Islam *et al.* [8] are suggesting the framework that allows software engineers to accurately elicit rather abstract regulatory requirements in the design stages of the development process and later ensure that these requirements are implemented correctly in following phases. Authors propose an interesting goal-modelling approach for eliciting security requirements using a security-oriented superset of *i** framework - Secure Tropos modelling language. While GDPR pays great attention to the security of information system the main corner-stones of the regulation are rather legal or "social" aspects and constraints of regulation (*e.g.* the "Consent" concept, when the owner of personal data can decide whether its data can be processed or not). Islam *et al*. are supporting modelling of legal constraints with help of the UMLsec modelling language and Model-Based Security Engineering (MBSE) approach. Later on, authors show few examples of how their approach can be used to model constraints proposed by Directive 95/46/EC [3] (GDPR predecessor).

However, authors explicitly notice that security requirements have to be defined in early stages of the development process, while the analysis of already developed information system does not get attention at all. The GDPR enforcement would still have a great impact on already developed information systems, thus the analysis of business processes of such information systems is very important, which this work is trying to achieve.

### 2.1.4 Adaptive and Runtime Compliance with Regulations

We found a set of studies that are discussing the so-called adaptive compliances [9] [10]. All studies in this block are sharing the same idea of business process adaptiveness to constant evolutions of regulations. This is possible due to monitoring of business processes during the software system runtime.

Ly *et al*. [9] conducted the systematic literature review of methods and frameworks of compliance monitoring and in support of their work, the unified framework for Compliance Monitoring Functionalities (CMFs) was proposed. Ly *et al*. remark, that study focus lies in the detection and prediction of compliance violations and activation of countermeasures of compliance violations. Authors state that this framework would allow to model different constraints derived from rather abstractly defined regulations and monitor the businesses processes during its execution. Furthermore, authors show how proposed framework can be used in three different compliance monitoring tools.

Garcia-Galan *et al.* [10] are proposing a seven-step process for adaptive compliance of business process in cloud computing services, where software should comply with different regulations because of its globalisation nature. Authors see the adaptive compliance as the only possible solution to systems that are operating in heterogeneous, highly distributed and dynamic environments. However, authors acknowledge limitations in the automation of majority of steps of the proposed process, due to current limitations in technology and most business processes nature, where human involvement is crucial nowadays. Besides this, Garcia-Galan *et al.* admitted that study overlooked questions related to the performance effects on the business process if such techniques would have been used.

Although the idea of adaptive compliance is the next step in the software system compliance field, still there are several challenges to solve. The regulations often come as an abstract and vague set of constraints and the main problem lies in the correct interpretation and derived implementation of business process constraints, which still, has to be done manually.

## 2.2 General Data Protection Regulation

As an answer to arising challenges in the field of personal data usage of the EU citizens the European Parliament and Council began work on the new legislation in the early 2010s [11]. The new regulation would standardize the rules of processing the personal data for all member states of the EU. GDPR will replace the current and outdated Directive 95/46/EC [3], which was implemented differently in the Member States, producing different interpretations of the law and leaving loopholes in the usage of personal data by organizations [12]. However, the new regulation will preserve the key points and definitions used in the old legislation, adding new concepts and unifying personal data protection principles [12]. The final edition of the legislation was approved in the April 2016 and will take effect at the end of the May 2018, giving organizations the 2-year transitional period [1].

The regulation application scope is defined in two articles [Art. 2] “Material scope” and [Art. 3] “Territorial scope” [1].

According to the territorial scope of the regulation, it applies to information systems that are processing personal data of EU citizens, even if the processing is done outside of the EU [1] [13]. Meaning that each IS that provides any service related to processing personal data of EU citizen should comply with GDPR.

The material scope defines the means and scope of the processing terms. The regulation applies if personal data is being processed using automated or manual processes [1]. According to Voigt and von dem Bussche [14], regulation [1][Art. 4 (2)] gives a very broad

definition of processing, giving a long list of the procedures that can be considered as the processing of the data, however, in the general processing means any manipulations performed with data.

## 2.3 GDPR Meta-model

In this subsection, we are analysing the legislation text and other sources [1] [14]. We discuss the definitions of the key aspects given by regulation to support our abstraction of GDPR in the shape of the meta-model. Later this meta-model can be used as reference point to check compliance of IS business processes 3.1. From the analysis of the legislation text, we derived a static meta-model of the GDPR. The proposed meta-model can be seen in Figure 1.

### 2.3.1 Personal Data, Data Subject and Filing System

The most important aspect in GDPR is personal data, and according to legislation: *"**'Personal data'** means any information relating to an identified or identifiable natural person ('**data subject**') …such as a name, an identification number, location data, an online identifier…"* [Art. 4(1)].

It is crucial to differentiate whether the processed data is personal or not since regulation applies only with the processing of the personal data [1]. Voigt *et al.* [14] suggesting that data becomes personal when it can be used to identify data holder (data subject), moreover if different parts of processed data can be combined to identify data subject (even using different sources), then this data should be also considered as personal [14]. Regulation gives examples of which data can be considered personal – name, identification number, an online identifier. Furthermore, regulation explains the concept of the online identifier in [Recital 30], where: *"Natural persons may be associated with online identifiers provided by their devices, applications, tools and protocols, such as internet protocol addresses, cookie identifiers or other identifiers such as radio frequency identification tags.",* similar to this, Voigt and von dem Bussche [14] are also suggesting that online identifier could mean, for example, an IP address or browser cookies.

In addition to that, GDPR defines different categories of personal data such as genetic, biometric and data concerning health [Art. 4 (13, 14, and 15)], furthermore in [Art. 9(1)], regulation defines special categories of data, that are prohibited to process for sole purpose of identification data subject, unless some exceptions [1]: *"Processing of **personal data** revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation shall be prohibited."* [Art. 9(1)].

According to regulation [1], personal data can be contained in filing systems: *"**'filing system'** means any structured set of personal data which are accessible according to specific criteria, whether centralised, decentralised or dispersed on a functional or geographical basis;"* [Art. 4(6)]; and *"This Regulation applies to the processing of personal data wholly or partly by automated means and to the processing other than by automated means of personal data which form part of a **filing system** or are intended to form part of a **filing system**"* [Art. 2(1)]. From these definitions, we can assume that filing system is a technology-based system, which can aggregate data by some common parameter, in other words, it is information system. However, according to Voigt *et. al.* [14], filing system should not be necessarily considered as a technology-based system, it could be for example, alphabetically organised set of documents in files.

To capture all this in our meta-model three classes are defined:

*(i)* **Personal Data**, with attribute category enumeration type (DATA_CATEGORY), attribute to help classify data (*e.g.* GENETIC);
*(ii)* **Data Subject**, the representation of data owner, who is associated with personal data with *owns* association.
*(iii)* **Filing System**, in our case it is a technology-based system, which contains, interacts and/or aggregates sets of data by specifically provided rules. Filing system directly associates with personal data with a *set of* association.

### 2.3.2 Actors Linked to Personal Data

GDPR defines a handful of actors that are directly or indirectly related to the processing of personal data whether they are working with data, determine the purposes of data processing, consume data or supervising these processes. For all of these actors' regulation starts its definition with *"means the natural or legal person, public authority, agency or other body"*. This similarity can be captured in one generalized class:

*(i)* **Authority,** generalization class that binds all actor classes together. Has enumeration attribute type (AUTHORITY_TYPE) to help determine the type of actor (*e.g.* PUBLIC).

Regulation determines actor that is responsible for defining the purposes of processing data: *"**'controller'** means the natural or legal person, public authority, agency or other body which, **alone or jointly with others**, determines the purposes and means of the processing of personal data…"* [Art. 4(7)].
From this definition, we can assume several things: *(i)* controller is an actor, who defines the purposes and means of processing data; *(ii)* there can be one or several controllers.
Furthermore, GDPR gives definition to a different type of authorities - actors that process personal data: *"**'processor'** […] body which processes personal data on behalf of the controller…"* [Art. 4(8)]; and *"**'third party'** […] body other than the data subject, controller, processor and persons who, under the direct authority of the controller or processor, are authorised to process personal data"* [Art. 4(10)].
From these definitions, handful of useful information can be extracted: *(i)* in some extension controllers can transit their authority to process personal data to other actors such as processor or third party; *(ii)* both, processor and third party, can process personal data; *(iii)* third party is a different actor and should be separated from controller and processor; *(iv)* there can be several processors.
Regulation mentions other actors that could access personal data through the processing activities of the controllers or processors, without actually processing it: *"**'recipient'** […] body to which the personal data are **disclosed** […]"* [Art. 4(9)]. The disclosure of personal data is mentioned in other articles of regulation, for example: *"'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;"* [Art. 4(12)] and *"[…] (3) The controller shall provide the information referred to in paragraphs 1 and 2:*
*[…] c) if a **disclosure to another recipient is envisaged**, at the latest when the personal data are first disclosed. […]"* [Art. 14(3c)]. From which we can conclude, that unauthorized disclosures should be considered as data breach which often cannot be planned, whereas the possible disclosures should be envisaged and information about them should be captured in advance.
The regulation defines two main subjects that will be dealing with the data – controller and processor. The controller is any authority or organization that is determining the purposes of processing the personal data, while the processor is the one who will be authorized by the

controller to process the data. Besides this, GDPR describes several additional actors that are dealing with personal data – recipient and third party. The third party can be authorized by processor or controller to process data, while the recipient is an actor for whom the data can be disclosed.
According to Voigt *et al.* [14], controller and processors can be separated by the decision-making power attributes. The controller has the power to make decisions about how data has to be processed (has greater responsibilities) and can authorize other processors to operate with personal data in some distinct. However, the regulation does not prohibit controller to process personal data by itself, whereas the third party cannot be any of processor or controller. From these definitions, these classes and relations can be defined:

*(i)* **Controller,** is a subclass of **Authority**, *determines* the **Purpose** of processing**.**
*(ii)* **Purpose,** class that defines the purposes or means to process data. The purpose has *to process* association with **Personal Data** class.
*(iii)* **Processor OR Third Party,** a class that is representing processor or third party, is associated with **Controller** class with *authorizes* association. Processors and third parties are using **Purposes** and **Filing System** to process **Personal Data**.
*(iv)* **Recipient,** a subclass of **Authority,** is connected to **Personal Data** class with associative class **Disclosure**.
*(v)* **Disclosure,** class that represents the disclosure of personal data.

### 2.3.3 Processing

The scope of the regulation explicitly states that regulation applies only if the processing of the personal data appears, which makes processing one of the key aspects of the GDPR and our meta-model. *"[…] **'processing'** means any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means such as collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use […]"* [Art. 4(2)].
Regulation gives a very broad definition of processing, providing a long list of the procedures that can be considered as the processing of the data, however in the general processing means any manipulations performed with personal data [14]. In addition to processing legislation text mentions special cases of processing such as: *"**'profiling'** means any form of automated processing of personal data consisting of the use of personal data to evaluate certain personal aspects relating to a natural person […]"* [Art. 4(4)]; *"**'pseudonymisation'** means the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information […]"* [Art. 4(5)]. *"**'cross-border processing'** means either:*

*a) processing…which takes place in the context of the activities of establishments in more than one Member State […], or*
*b) processing of personal data […] which substantially affects or is likely to substantially affect data subjects in more than one Member State"* [Art. 4(23)].

Besides this, in [Art. 30(1, 2)] GDPR mentions the responsibility of controller and processor to collect records of the processing: *"Each controller/processor and, where applicable, the controller's/processor's representative, shall maintain a record of processing activities under its responsibility."*. The record of processing shall contain general information such as controllers/processors contact details, purposes of processing, links to personal data and data subject [Art. 30]. The records can be kept digitally and should be made available to supervisory authority on request: "*The records referred to in paragraphs 1 and 2 shall be in writing, including in electronic form.*" and "*The controller or the processor and, where*

*applicable, the controller's or the processor's representative, shall make the record available to the supervisory authority on request.*" [Art. 30(3, 4)].
To capture processing activities, we added next classes to our meta-model:

*(i)* **Processing -** associative class between the **Processors OR Third Party** and **Personal Data** classes. Processing has two attributes: operation (PROCESSING_OPERATION) – enumeration type (e.g. COLLECTION or PROFILING) and the Boolean type attribute cross-border, which would help to determine whether processing is cross-border or not.

*(ii)* **Record** – associative class between **Processing** and **Personal Data,** it is connected with *logs* association.

### 2.3.4 Consent

Alongside the personal data, the consent is one of the key aspects of the regulation: *"**'consent'** of the data subject means any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her;"* [Art. 4(11)].
Prior the data processing the controller should obtain the clear indication of agreement for each specific purpose of data processing from data subject: *"[…] the request for consent shall be presented in a manner which is clearly distinguishable from the other matters, in an intelligible and easily accessible form, using clear and plain language […]"* [Art. 7(2)]. The agreement should be obtained by a clear affirmative action, which means data subject should make an action to give consent (e.g. signify the agreement by reading it and clicking a button or checking checkbox) [13] [Recital 32]. The consent can be given to one or several purposes: *"[…] the data subject has given consent to the processing of his or her personal data for one or more specific purposes;"* [Art. 6 (1a)]. As well as that, GDPR Art. 7 mentions that: *"[…] Where processing is based on consent, the controller shall be able to demonstrate that the data subject has consented to processing of his or her personal data […]"* [Art. 7(1)] and *"[…] Prior to giving consent, the data subject shall be informed thereof. It shall be as easy to withdraw as to give consent. […]"* [Art. 7(3)].
From here the **Consent** class can be defined, it is associated with **Data Subject** class with *signifies* association and associative class **Agreement**. **Consent** is associated with **Purpose** class with *given for* relationship. These classes would help to capture the concept of consent, which is given for each purpose by the data subject. The **Agreement** class would help to prove the existence of the consent if later the consent for processing should be requested by supervisory authorities. To mark whether the consent was legally collected next Boolean type arguments can be added to **Consent** class: *is_freely_given, is_specific, is_informed_of_withdrawal, is_unambigous, given_with_affirmative_action, is_clearly_distinguishable*. The purpose and usage of these attributes will be explained later on in 2.3.6.
Nonetheless, the legislation draws attention to some exclusions that can free controllers or processors from gathering consents from data subject:

*(i)* *"[…] processing is necessary for the performance of a contract to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract; […]"* [Art. 6(1b)]. In other words, if data subject and processor are willing to enter into contract defined relationship [Recital 44]. To help capture this, to **Purpose** class was added a Boolean-type attribute *contract_processing*.

(ii) *"[...] processing is necessary for compliance with a legal obligation to which the controller is subject; […]"* [Art. 6(1c)]. In other words, if there is a legal basis that allows the processor or controller to process personal data without the consent of data subject [Recital 45]. To capture that to **Purpose** class was added Boolean-type attribute *legal_obligation.*

(iii) *"[…] processing is necessary in order to protect the vital interests of the data subject or of another natural person;"* [Art. 6(1d)]. To capture that to **Purpose** class was added Boolean-type attribute *vital_interests.*

(iv) *"[…] processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller; […] "*[Art. 6(1e)]. To capture that to **Purpose** class was added Boolean-type attribute *public_interest.*

(v) *"[…] processing is necessary for the purposes of the legitimate interests pursued by the controller or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, in particular where the data subject is a child. […]"* [Art. 6(1f)]. To capture that to **Purpose** class was added Boolean-type attribute *legitimate_interest.*

(vi) *"[...] If the purposes for which a controller processes personal data do not or do no longer require the identification of a data subject by the controller, the controller shall not be obliged to maintain, acquire or process additional information in order to identify the data subject for the sole purpose of complying with this Regulation. [...]"* [Art. 11(1)]. To class **Purpose,** Boolean attribute do_not_require_identification was added.

### 2.3.5 Rights of Data Subject

GDPR explicitly defines a handful of rights of data subject when the personal data is being processed. In this part of the section, we are reviewing rights that can be implemented in the information system (*i.e.* rights that can be defined using business process notation and, in our opinion, can be integrated into information system). The decision of whether the certain right is applicable in given situation or not is in the scope of this analysis, thus we only acknowledge that these rules have to be implemented in observable IS, and decision-making power should be transferred to the appropriate authority in this case.
To generalize rights, the abstract class "**Right to"** can be defined, which is associated with **Data Subject** class with *has* association. Next rights are being discussed:

*(i)* Right of Access by the Data Subject

*"[…] the data subject shall have the right to obtain from the controller confirmation as to whether or not personal data concerning him or her are being processed […]"* [Art. 15(1)]. According to Voigt and von dem Bussche [14], the right has to be implemented in two steps. Firstly, the data subject has right to ask whether the processing of its personal data takes place. Secondly data subject can gain access to additional information (purposes of processing, categories of personal data concerned, recipients or categories of recipient to whom the personal data have been or will be disclosed) if the processing of his/her personal data takes place. To capture this in GDPR meta-model, class **Access** is added and associated with class Processing with association *allows to get information about*.

*(ii)* Right to Rectification

*"[…] the data subject shall have the right to obtain from the controller without undue delay the rectification of inaccurate personal data concerning him or her. Taking into account the*

*purposes of the processing, the data subject shall have the right to have incomplete personal data completed, including by means of providing a supplementary statement. […]"* [Art. 16]. That would mean, that there should be implemented a mechanism that would allow editing falsely documented personal data. However, Voigt *et al.* [14] mention that the burden of proof showing that personal data is misrepresented should be carried by the data subject in this case. This right is captured with class **Rectify** and it is associated with **Personal Data** class with association *allows to correct*.

*(iii)* Right to Erasure (Right to be forgotten)

*"[…] The data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay […]"* and *"[…] Where the controller has made the personal data public and is obliged pursuant to paragraph 1 to erase the personal data […]"* [Art. 17(1, 2)]. There should be a mechanism that would allow removing certain sets of personal data if the data subject is willing so. However, there are some exclusions of when the controller can deny the request for personal data erasure [Art. 17(3)]. To capture this right the class **Erasure** is added to meta-model, it is associated with **Personal Data** class with association *allows to erase*.

*(iv)* Right to Restriction of Processing

The right to the restriction of processing is defined across the regulation in several articles: *"[…] 'restriction of processing' means the marking of stored personal data with the aim of limiting their processing in the future […]"* [Art. 4(3)]; *"[…] The data subject shall have the right to obtain from the controller restriction of processing […]"* and *"[…] Where processing has been restricted under paragraph 1, such personal data shall, with the exception of storage, only be processed with the data subject's consent […]"* [Art. 18(1, 2)]; *"[…] The data subject shall have the right to withdraw his or her consent at any time. The withdrawal of consent shall not affect the lawfulness of processing based on consent before its withdrawal […]"* [Art. 7(3)]. This means that data subject has right to withdraw the previously given consent and restrict the processing of the personal data. In this case, the personal data should be marked and not processed further. This right is represented as **Restrict Processing** class with association *allows to withdraw* to **Consent** class and *allows to limit* the association to **Processing** class.

*(v)* Right to Data Portability

*"[…] The data subject shall have the right to receive the personal data concerning him or her, which he or she has provided to a controller, in a structured, commonly used and machine-readable format and have the right to transmit those data to another controller without hindrance from the controller to which the personal data have been provided […]"* and *"[…] In exercising his or her right to data portability pursuant to paragraph 1, the data subject shall have the right to have the personal data transmitted directly from one controller to another, where technically feasible […]"* [Art. 20(1, 2)]. Data subject should be able to transform collected personal data from one controller to another in commonly spread data format. According to Voigt *et al.* [14] legislation is mainly targeting social networks in this situation, however, there can be other examples as well. To represent this right **Portability** class was added to meta-model and associated with **Personal Data** with *allows export of* association.

*(vi)* Right of Notification

*"[…] the controller shall communicate any rectification or erasure of personal data or restriction of processing carried out in accordance with Article 16, Article 17(1) and Article*

*18 to each recipient to whom the personal data have been disclosed, unless this proves impossible or involves disproportionate effort. The controller shall inform the data subject about those recipients if the data subject requests it. […]"* [Art. 19]. This right is being triggered by three other rights: erasure, restriction and rectification. When personal data was disclosed to third parties' controller should notify data subject of each erasure, restriction and rectification of personal data carried by third parties. This right is represented as Notification class with associations *triggered* to **Erase**, **Rectify**, **Restrict Processing** classes. In addition, it is associated with Disclosure class with *allows to get information* association.

### 2.3.6 Dynamical Constraints

Static class meta-model alone does not allow us to capture cases when the application of regulation can be dynamically changed. In order to capture such conditions, we are proposing a set of constraints, which will help dynamically change the collection of compliance classes and define additional compliance rules in our meta-model, see Table 1.

Table 1. Dynamical constraints for meta-model

| ID | Constraint Definition |
|---|---|
| C1 | *Consent is valid only when all its attribute Boolean fields are TRUE* |
| C2 | *Consent can be given by Data Subject only if age is > 15, otherwise by its representative.* |
| C3 | *If any of Boolean field of Purpose class is TRUE, then Consent is not needed to be collected.* |
| C4 | *If Boolean attribute do_not_require_identification of Purpose class is TRUE, then, rights of the data subjects do not have to be applied.* |
| C5 | *Only authority (Controller) with type OFFICIAL can authorize processing of Personal Data with CRIMINAL category.* |

**C1.** The regulation defines set of rules to be followed when consent is being collected from the data subject, which we covered in Chapter 2.3.4. The application of these rules is represented as Boolean type arguments of class Consent, if any of the rules are successfully implemented then it should be positively marked under appropriate attribute. Only if all these rules are considered to be applied correctly, then consent should be marked as correctly compiled.

**C2.** It is crucial to mention that the consent cannot be gathered directly from a data subject when he/she is younger than 16 years, in this case the consent from holder of parental responsibility should be asked: *"[...] the processing of the personal data of a child shall be lawful where the child is at least 16 years old. Where the child is below the age of 16 years, such processing shall be lawful only if and to the extent that consent is given or authorised by the holder of parental responsibility over the child. […]"* [Art. 8(1)]. If this constraint fails, then consent should not be considered as correctly collected, and therefore to be invalid.

**C3.** GDPR lists some exclusions of when consent to process personal data is not needed to be collected. We covered these situations in the previous part of this chapter - 2.3.4**.** Similarly, to the C1 application of these situations is described using Boolean type's attributes in Purpose class. However, now if at least one of these attributes is evaluated positively, then it elevates obligation of collecting consent to process personal data from processors.

**C4.** The regulation provides a situation, when the rights of the data subject can be neglected while processing his or her personal data, from [Art. 11(2)]: *"[...] where, in cases [...] the controller is able to demonstrate that it is not in a position to identify the data subject, the controller shall inform the data subject accordingly, if possible. In such cases, Articles 15 to 20 shall not apply except where the data subject, for the purpose of exercising his or her rights under those articles, provides additional information enabling his or her identification."* Meaning, if there is no possibility to identify data subject based on personal data at hand, then the rights mentioned in Chapter 2.3.5 should not be implemented.
**C5.** Another very specific article "Processing of personal data relating to criminal convictions and offences" [Art. 10] of GDPR lays strict constraints on the processing of special category personal data. Namely, personal data relating to criminal convictions: *"Processing of personal data relating to criminal convictions and offences or related security measures based on Article 6(1) shall be carried out only under the control of official authority or when the processing is authorised by Union or Member State law providing for appropriate safeguards for the rights and freedoms of data subjects. Any comprehensive register of criminal convictions shall be kept only under the control of official authority."* [Art. 10(2)]. This allows us to define another dynamic constraint since the category of personal data and authority types can change. For this constraint, we will look for personal data with category CRIMINAL, and check whether the controller has type OFFICIAL since only processing of this personal data should be under control of the official authorities.

## 2.4 Summary

In this chapter, we gave a short overview of a GDPR and analysed the legislation text, which supported the design of proposed meta-model. This meta-model is an abstraction of the regulation from the perspective of the IS business processes compliancy towards GDPR. Meta-model consists of two main parts: *(i)* static class model; and *(ii)* dynamical constraints. The static model helps to define main actors and entities that are taking part (or not taking) in the business process, while dynamical constraints allow us to capture different scenarios of the business process. With this chapter we gave an answer to our first sub-question - SUBQ1, namely short overview gives the picture, of what this regulation is, and the meta-model is our way to formalise it.

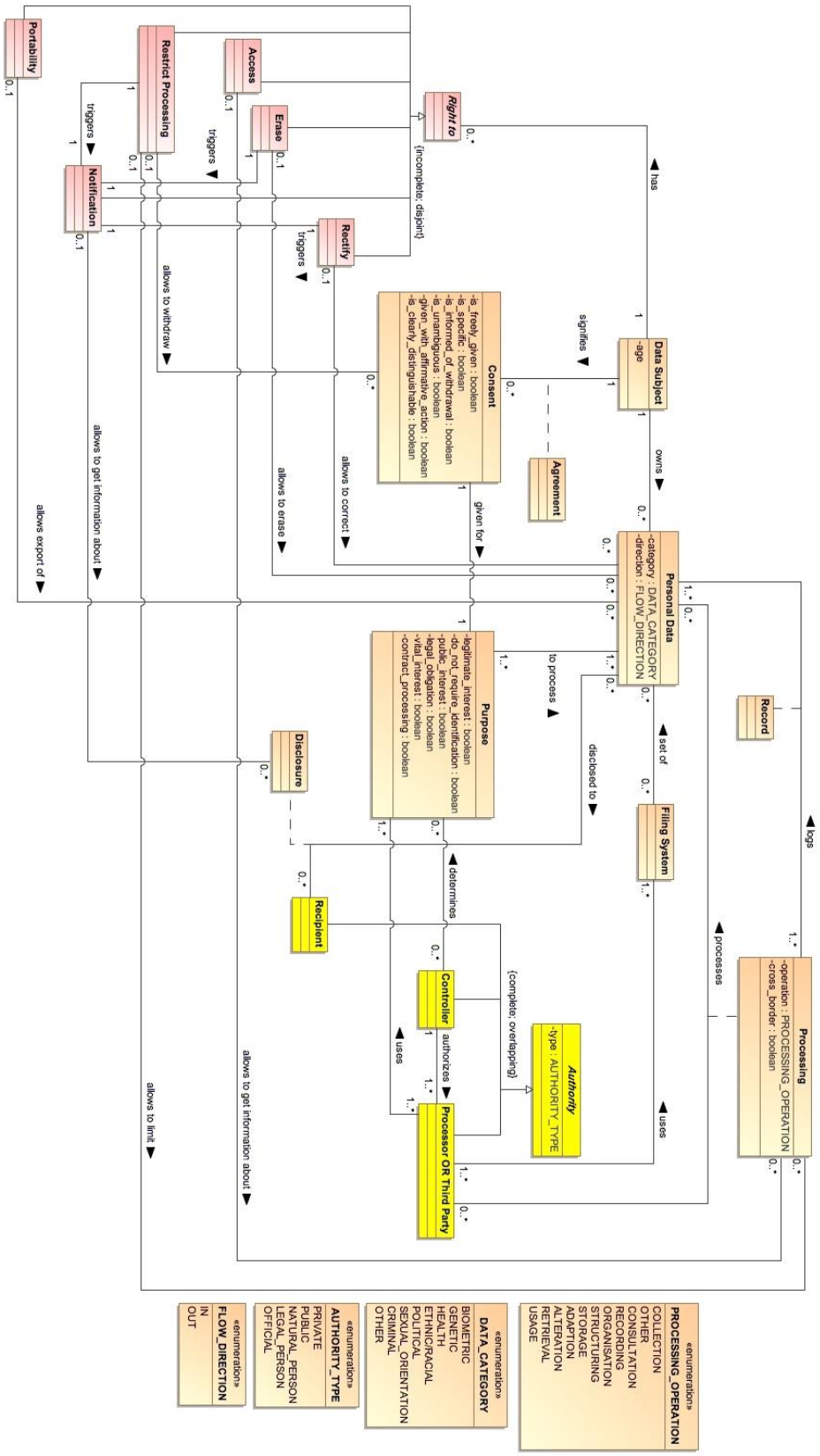


Figure 1. Proposed GDPR meta-model

# 3 Method of GDPR Compliance Analysis

Once we found answers to our first sub-question (SUBQ1) of our work, we can now proceed to our next sub-question – SUBQ2. This chapter is focusing on our main contribution to this work, namely, in this chapter we are presenting our method of checking business processes compliancy with GDPR, moreover, we support method with a use case example of business process derived from developing IS.

## 3.1 Method

In this sub-section, the method of checking compliance of business processes towards GDPR is presented. This method is iterative and can be performed several times. The high-level steps of the method are depicted in Figure 2:

I. **Extract as-is compliance model.** This step will take the actual BPMN models of the business processes and additional input from the user to help transform BPMN model to UML notation as-is compliancy model.

II. **Compare two meta-models.** Once as-is compliance model of all relevant (the processes that directly or indirectly deal with processing of personal data) business processes is instantiated it can be compared with previously defined GDPR meta-model.

III. **Define compliance issues.** Based on found differences of two models this step will give a binary answer to question whether extracted compliance model is GDPR-compliant or not, moreover, this step would give detailed descriptions of business process incompliances with GDPR.

IV. **Change business processes models.** This step is optional and should be taken if the output of the previous step is not satisfactory. The input for this step would be generated report of compliancy issues of business process that would help to change it.

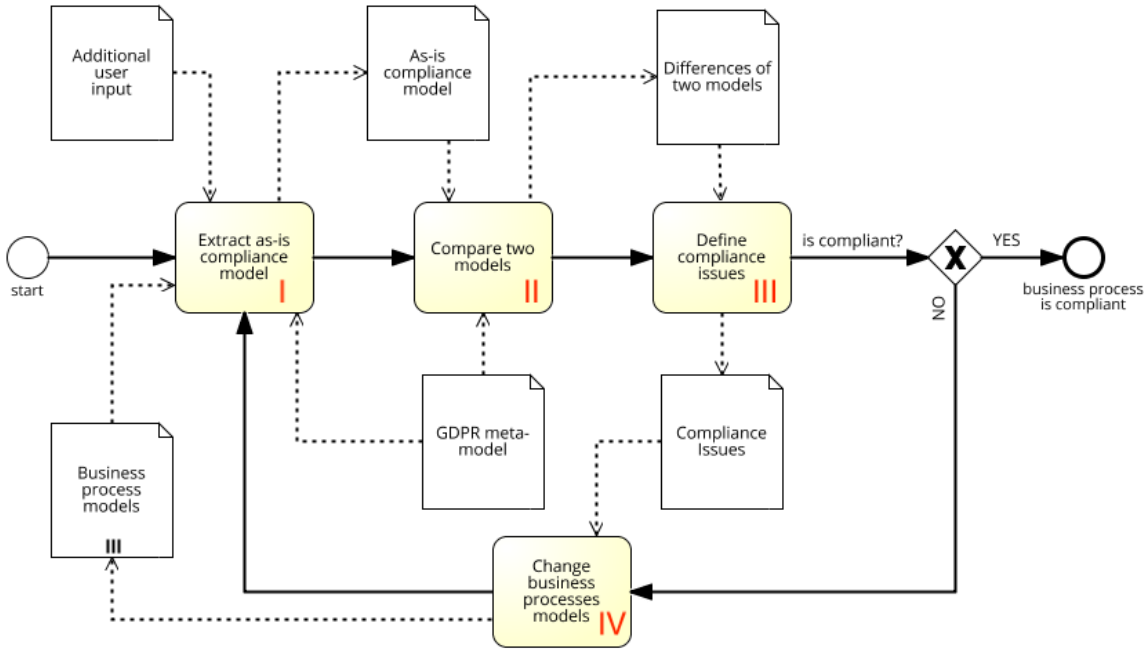


Figure 2. Method of GDPR Compliance Analysis

In the next sub-sections these high-level steps are discussed in detail, moreover to illustrate each step, three business processes derived from analysis of real use case IS are being taken as running example.

## 3.2 Running Example

The information system used as running example is called ÕIS2[2] (Haridustasemete ülese õppeinfosüsteem 2). ÕIS2 is developed by Fujitsu Estonia AS[3], procured by Estonian Educational Technology Foundation[4] (HITSA) and funded by European Structural and Investment Funds[5]. The developing IS will serve as study information system for Estonian colleges, vocational schools and professional higher education institutions. We picked this information system as running example because of the amount of the personal data processed in this system. The information system will process a handful of different personal data, such as students' academic performance, students and parents contact information and in some extend well-being of students (sick leaves, students home situation, students special needs if any *etc.*). In order to illustrate our methodology, we conducted a short analysis of this use case information system and handpicked three business processes from original analysis of ÕIS2, the results of such analysis can be found in the Appendix to this thesis: Running Example Analysis.

## 3.3 Extraction Rules

This sub-section discusses proposed method's first step in detail (Extract as-is compliance model, Figure 2). In order to continue with the analysis of observable IS we have to define as-is compliancy model of the provided business processes. To accomplish that, useful information has to be extracted from one or several business process models using extraction rules defined below. Solely for the purpose of illustration, we slightly changed one of the observable processes to support all variations of the extraction rules. We will use changed process (see Figure 4) instead of the original process (see Figure 30), which was derived from analysis of the running example. Next changes have been applied to original process:

*(i)* Prior to task "Create new user" {8} we added activity (see Figure 3) that would ask for consent from the user ("Ask for Consent" {5}). Now the process is designed this way that it would not proceed with processing of personal data before the consent for it has been given. The activity of asking for consent can be seen in Figure 3, where ÕIS2 asks User for consent {1}, then waits for a response. When the response is gathered, the system can decide whether consent was given or not. Note that, if positive scenario takes place, then the agreement would be stored in the system.

*(ii)* After each processing activity "Check user ID code" and "Create new user" {3, 8} we added two activities, that would record processing of personal data – "Log processing" {7a}. This would allow us to illustrate the existence of the Record class in our meta-model.

*(iii)* To illustrate classes of Recipient and Disclosure we added a new pool to process: "External Recipient" {C}. Let's assume that after "Create new user" {8} activity ÕIS2 is disclosing some of the personal data to an external recipient, for statistical purposes, of how much unique users are registered in the system.

[2] https://projektid.hitsa.ee/display/IS2/Tahvel
[3] http://www.fujitsu.com/ee/
[4] http://www.hitsa.ee/about-us
[5] http://www.eib.org/products/blending/esif/index.htm

Definition of extraction rules is contributing to our major goal of meta-model-based method discovery. However, besides our ultimate goal, we are pursuing another minor goal, which is the possibility of method automation and delegation of compliance checks to the computer system. In our opinion, a major part of the rules can be automated to be used in computer systems. However, sometimes not all data can be extracted from business process models (*e.g.* it is difficult or impossible to determine controller or processor-based only on business process models), in this case, the additional information has to be collected from other sources (*e.g.* additional input from the user, modification of standard BPMN notation). Based on the user (here and onwards user of the method) involvement in the extraction process, we can define three types of rules:

*(i)* **Automatic –** rule can be applied without any user involvement in the extraction process and entities that will be extracted can be modelled using standard BPMN notation. *Note: All of the relationships between objects in as-is compliancy model are added automatically once target and source classes of the relationship are presented in the model.*

*(ii)* **Semi-automatic –** rule can be applied without user involvement into the direct process of extraction, however, prior to extraction, the user would have to apply some minor modifications at the process modelling stage (*e.g.* specifically marking activities to define the type of the represented class).

*(iii)* **Manual –** rule cannot be applied without user involvement in extraction process and information has to be provided from the context of a business process, which is not known or accessible by rule applier (computer system).

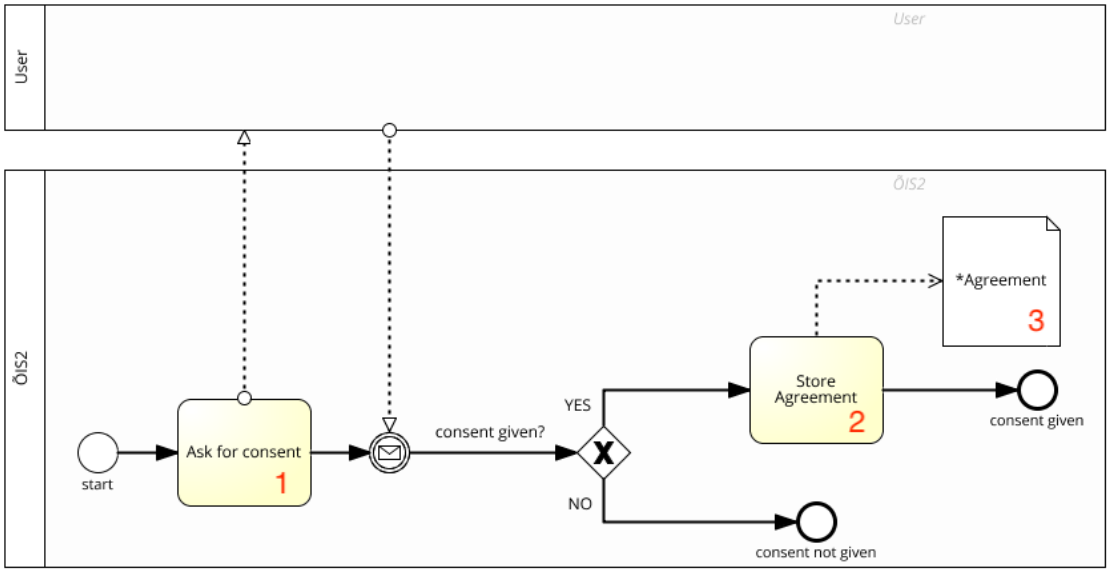


Figure 3. Ask for Consent sub-process

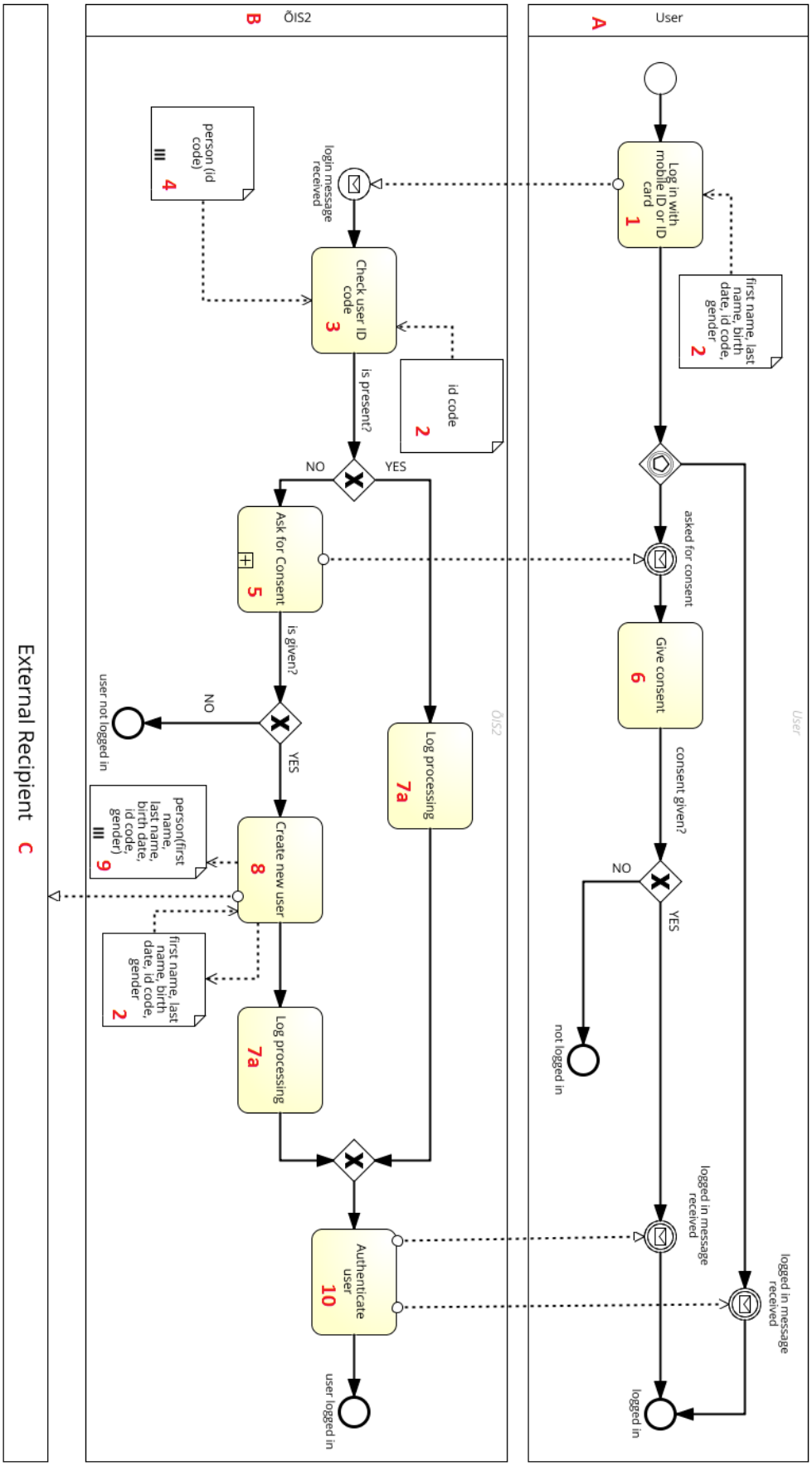


Figure 4. User Login Running Example

### 3.3.1 ER1. Controller, Processor and Third Party

The business model does not provide additional information on business process authority actors and such information has to be extracted from the context or meta-data of the process. That means that all information about Controller, Processor and/or Third-Party classes (types and titles) has to be provided by the user, hence type of this extraction rule is manual. ***Example**: Our example business models don't provide information about controller, processor and third-party entities. However, from the context provided above, we can conclude that in this case procurer of the software system (HITSA) is acting as both: Controller and Processor. There are several ways to represent this situation: see* Figure 5. *Controller and processor can be separated, or, both classes can be merged in one. Note that Controller and Processor classes inherit type attribute from the parent abstract class Authority (see* Figure 1*), in this case, we marked a type as PUBLIC.*

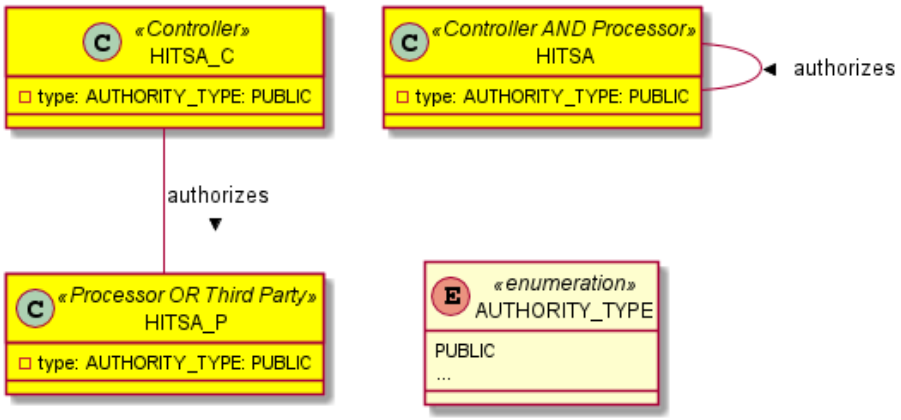


Figure 5. Example ER1 (i)

### 3.3.2 ER2. Personal Data and Data Subject

When extracting Personal Data class and Data Subject class's information, several sub-rules have to be considered:

*(i)* Personal data in BPMN can be represented as data objects, where instances of used Personal Data can be depicted as a list of strings separated by a comma. In this case, labels have to be extracted separately and each label has to be treated as one instance of Personal Data. To trace same personal data usage across one or several business processes, labels should be unique in regards of all IS observable business processes (*e.g.* personal data labelled as 'name' would be different from 'first name'). This rule can be performed automatically.
***Example**: Data object depicted in Figure 4 {2} (3 data objects with same data) is representing object of user's personal data with set of labels separated by commas, application of this rule will result in five different instances of Personal Data (with labels first name, last name, birth date, id code etc.) see Figure 6.*

*(ii)* Data objects can contain information of several Data Subjects. In this case, subjects have to be separated with line change and Data Subject label, and all Personal Data labels of a single Data Subject have to be contained in parenthesises. This rule can be automated.
***Example**: Data object depicted in Figure 32 {3} (3 Data objects with same data) represents personal data of two **Data Subjects** (Student and Parent) with Personal Data instances (id code, first name, last name etc.). This would tell us that activity is using personal data of two data subjects with different personal data sets, this way we can differentiate between several subjects (the result can be seen in Figure 7.)*

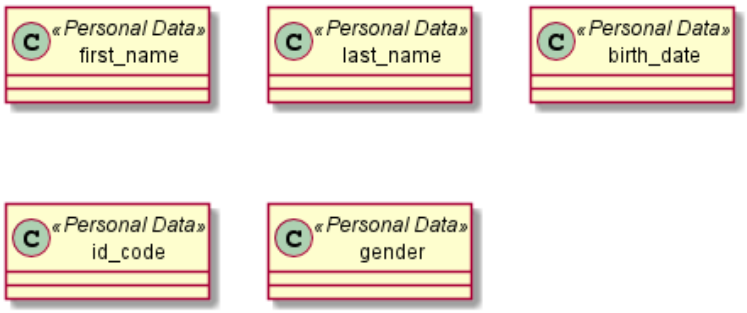


Figure 6. Example ER2 (i)

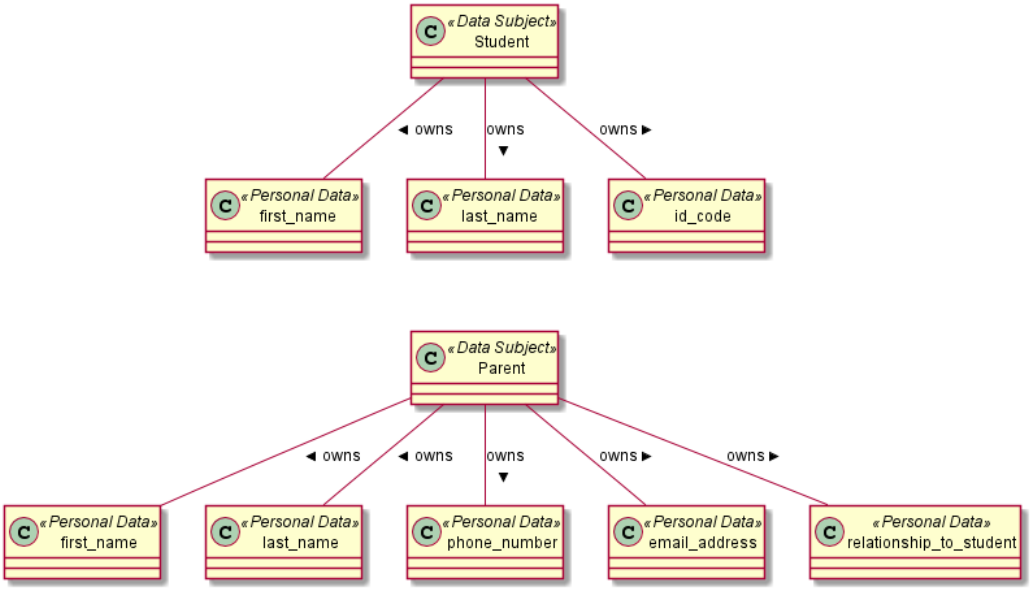


Figure 7. Example ER2 (ii)

*(iii)*If rule ER (i) is applied, then there is no annotated Data Subject in the data object, however, information about Data Subject can be extracted from the pool or lane labels where the personal data was used. It is complex to detect **Data Subject** this way, because the business process will typically have more than one lane/pool and thus in this case we have to detect Filing System object before that, hence extraction rule ER3 has to be applied before ER (iii). This rule can be implemented as automatic.

***Example:*** *See Figure 4 {A, 1, 2} and {B, 8, 2}, both actors 'User' and 'ÕIS2' are using the same set of Personal Data, however we can state that in this case pool with name 'User' acts as Data Subject, because it holds personal data and pool with name ÕIS2 is not Data Subject (see rule ER3 and result of the extraction can be seen in Figure 8).*

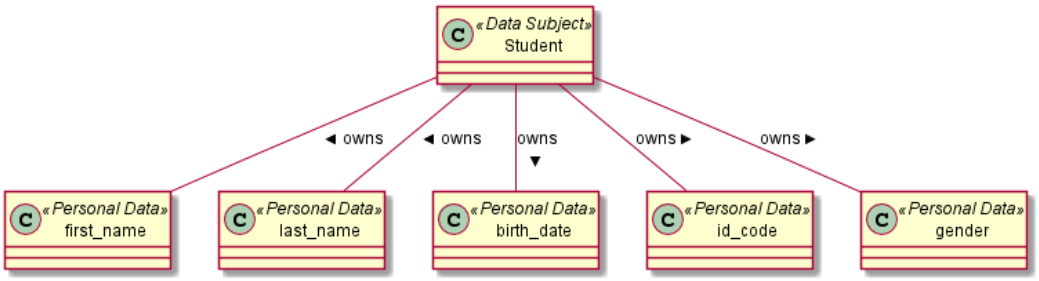


Figure 8. Example ER2 (iii)

*(iv)* The connection between the data object and task represents the flow of Personal Data, it can be ingoing and outgoing (If message flow arrow points towards data object – outgoing, if arrow points towards activity - ingoing). This information has to be noted, later it can be used to determine the flow direction of Personal Data between different parties and activities and define other classes such as Disclosure. This rule can be implemented automatically.
***Example**: Task 'Create new user' {8} has ingoing and outgoing connections with data objects {2, 9} (see Figure 4). In this case, we can assign two directions to one of the instances being used in this activity - id code, see Figure 9.*

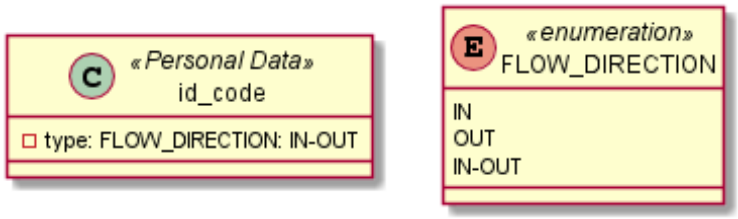


Figure 9. Example ER2 (iv)

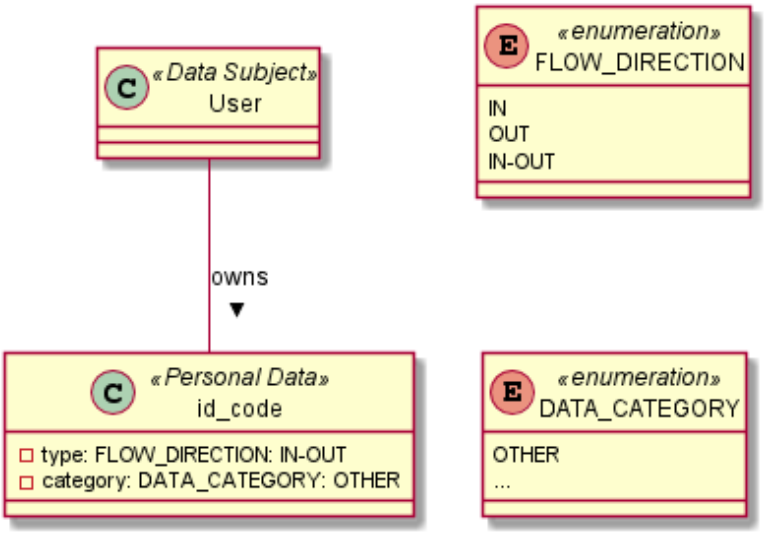


Figure 10. Example ER2 (vi)

*(v)* Personal Data can be contained in collections that are used by the information system to store information (*e.g.* database), to represent this, data object with property 'set' has to be used. If mark 'set' is being used, then different tables can be represented simultaneously in one data object, similarly as in ER2 (ii), where the same situation is used to define several data subjects. This rule is automatic.

***Example:*** *See Figure 4, data objects {4, 9} are sets they represent usage of database table person and Personal Data columns such as id code, first name, last name etc. That would mean, that activity processes data of not only one particular data subject, but rather it has access to several data subjects' personal data information.*

*(vi)* Our current set of extraction rules does not provide automation possibility to determine category attribute of Personal Data, hence it has to be defined manually.
***Example:*** *See Figure 4, data object {2} is representing usage of personal data such as id code, first name, last name etc. Since this data does not represent any category of listed in DATA_CATEGORY enumeration (Figure 1), other than OTHER then we can determine the category of data as OTHER. The result can be seen in Figure 10.*

### 3.3.3 ER3. Filing System

Filling System in the business process is usually represented as pool or lane. However, not all pools or lanes in observable processes are acting as a filing system. Pools with activities that have ingoing or outgoing connections to data objects with mark 'set' can help to identify **Filing System** class**.** If the information system is not operating with the database in the observable business process, then one of the pools has to be marked as **Filing System** manually. This rule can be applied automatically and manually.
From now on we will focus on one personal data instance (id code) for the more convenient perception of the extraction rules and will build our example as-is compliancy model around it. We added to our next example previously discovered entities of Processor and Controller, for better understanding how as-is compliancy is forming step-by-step.
***Example****: See Figure 4, ÕIS2 {B} is Filing System, because it has tasks 'Check user ID code' and 'Create new user' {3, 8} with ingoing or outgoing connections to data objects with mark 'set' {4, 9}. The example can be seen in Figure 11.*

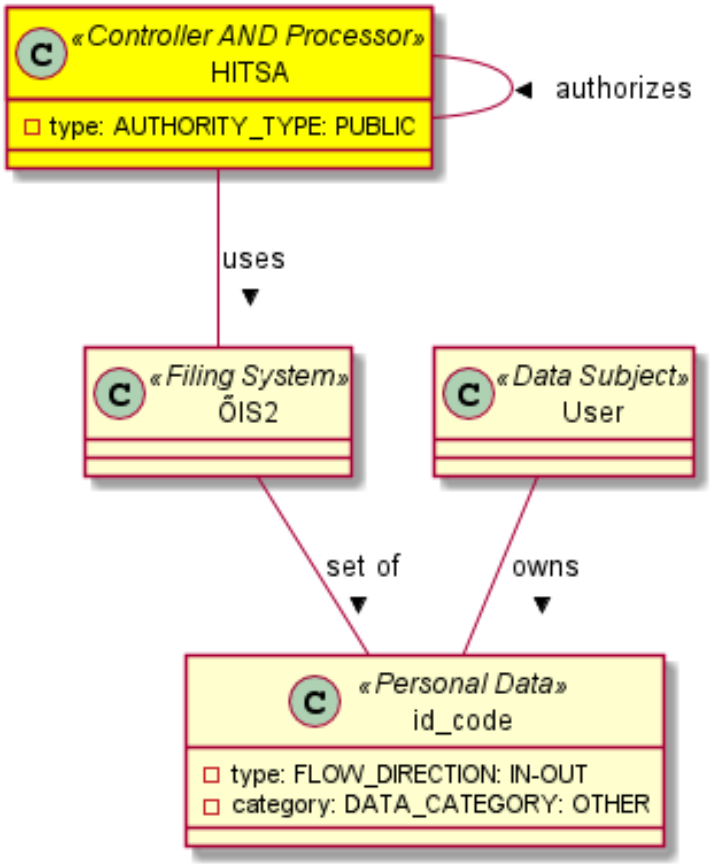


Figure 11. Example ER3

### 3.3.4 ER4. Processing

Processing is represented as activities of business process, with ingoing or outgoing connections to data objects in pools that represent Filing System**.** Processing attributes *operation* (via enumeration PROCESSING_OPERATION) and *cross_border* (via Boolean class) have to be defined manually by the user. The identification of processing object is automatic, and identification of its attributes is manual.

***Example:*** *See Figure 4, tasks 'Check user ID code' and 'Create new user' {3, 8} are Processing instances, because they have outgoing or ingoing connections with data objects {2, 4, 9}, and these tasks are in the pool of ÕIS2, which previously was identified as Filing System). Let's take only 'Create new user' task as an example and look at it more closely. From the context of the business process and name of the activity we can assume that this task is representing data persistence activity, moreover attached artefacts also suggest that data objects used in activity are exported to data collection {9}. In this case, we can conclude that this processing operation is STORAGE, and since nothing references to cross-border processing we will mark cross_border attribute as false (Figure 12).*

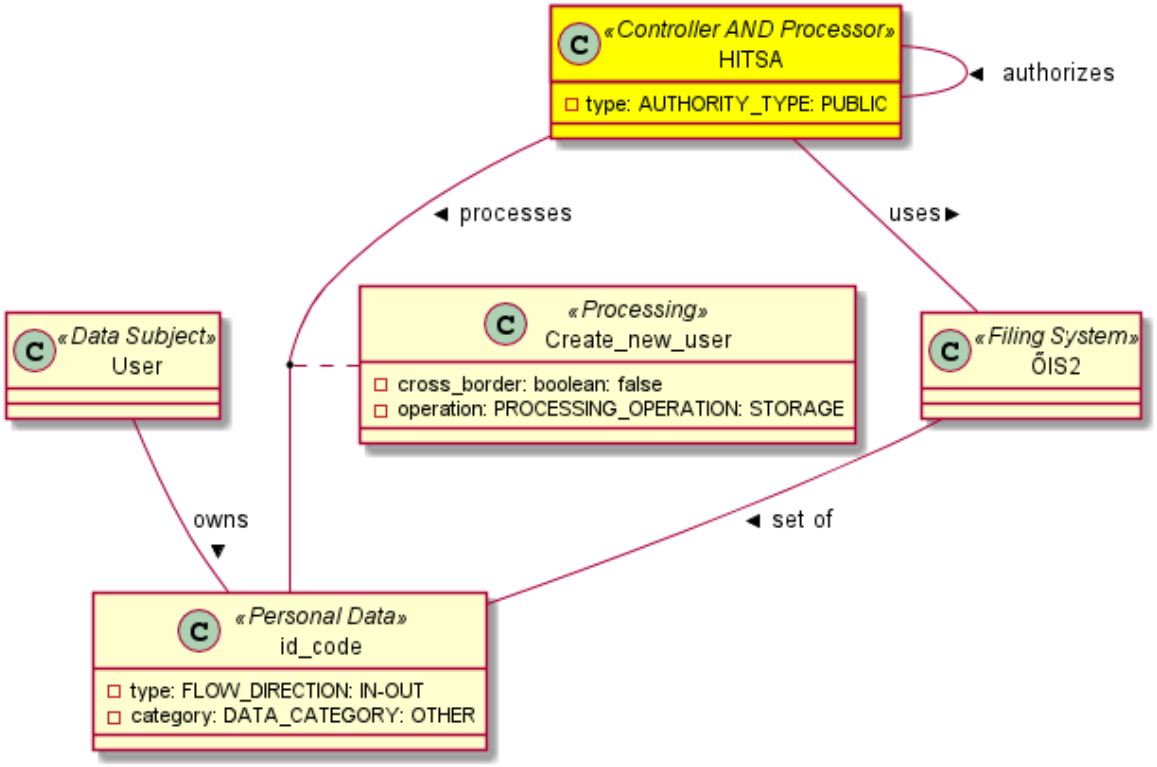


Figure 12. Example ER4

### 3.3.5 ER5. Record of Processing

Since it is only possible to record processing of Personal Data after processing activity was performed, recording task most likely will follow the original processing task. Hence the Record class can be represented by a special activity that is following each Processing task. However, the strategy of using only flow consecution to identify processing activities will be simply not enough to correctly identify them. Processing activity can be followed with other tasks, which are not related to logging. Hence in addition to flow consequence, we have to take into account the context of the tasks. To support automated detection of such task without using context, we used built-in functionality of BPMN 2.0 notation, to mark logging activities while designing process, namely, we added a property named 'logging' to each of such task because of that this rule can be classified as semi-automatic.

***Example**: Processing tasks 'Create new user' and 'Check User ID code' are followed by activity 'Log processing' (see Figure 4, {8, 3, 7a}). From the context of the tasks we can assume, that these are logging activities, the result can be seen in Figure 13.*

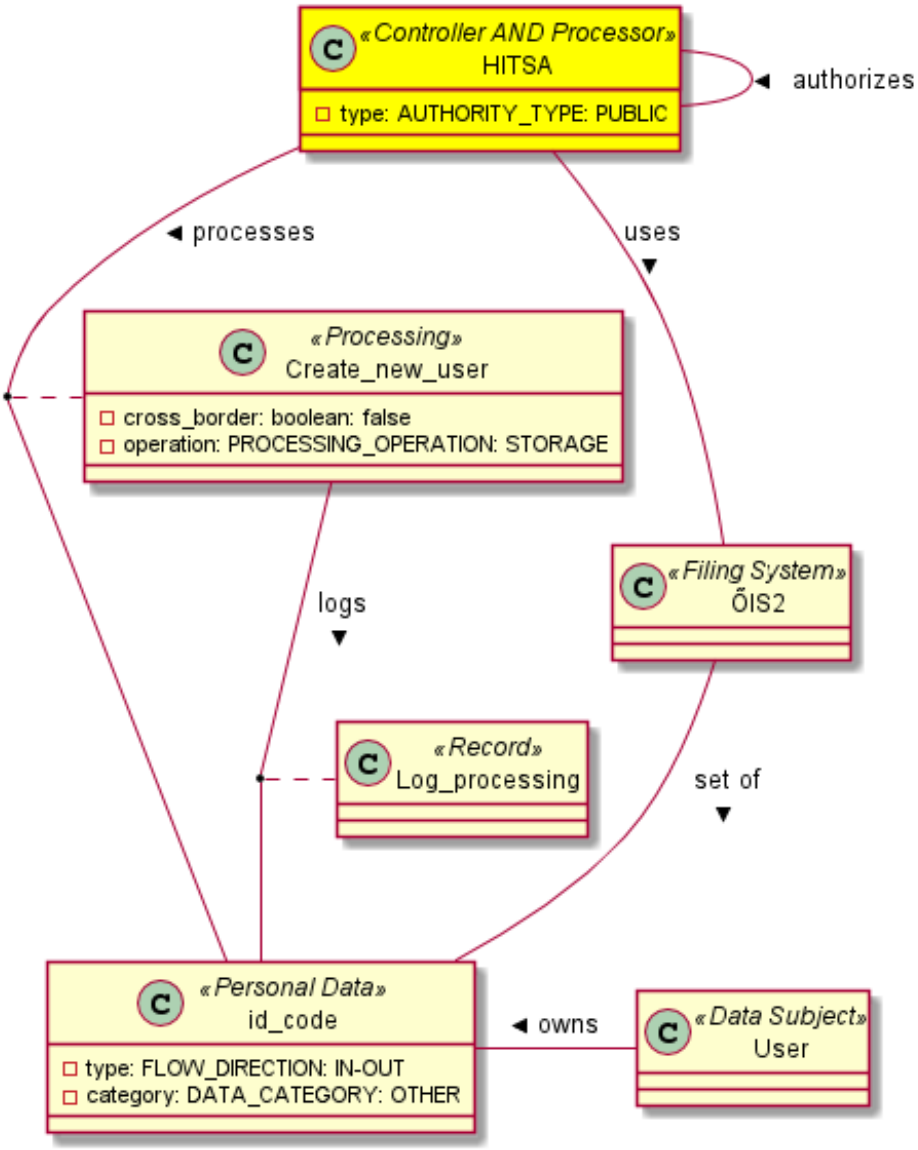


Figure 13. Example ER5

### 3.3.6 ER6. Purpose

Every instance of Purpose and all its attributes have to be added manually, the relations between Purpose class and other classes can be added automatically as soon as class is defined. Usually, one Purpose class is responsible for one Processing class, e.g. it defines the intention and rules of processing. Using our findings in Chapter 2.3.4 and context of the processing task, we can define the attributes of the Purpose class.

***Example**: 'Create new user' is processing task and uses personal data of User such as first name, last name, birth date etc., in this case, the Purpose class, which is responsible for this processing has to be added and linked with named Personal Data instances (see Figure 4, {8, 2}). Moreover, from the context of the task, all of the attributes can be defined as false since the processing does require identification of data subject, it is not a legitimate interest, and neither is it public interest, legal obligation, vital interest or contract processing. From*

*the context of the business process, we can conclude that purpose name is to store user information (see Figure 14).*

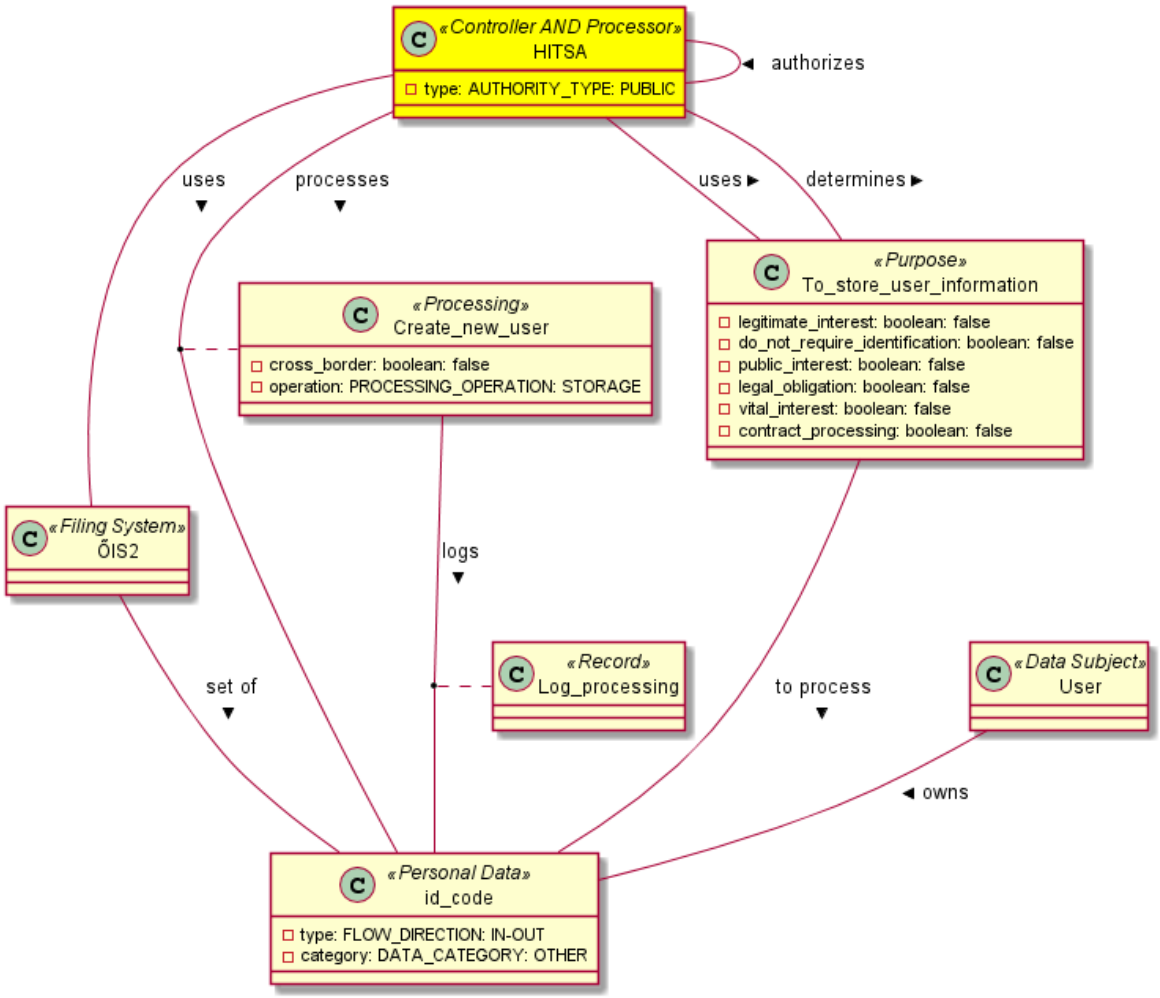


Figure 14. Example ER6

### 3.3.7 ER7. Consent and Agreement

Consent class can be represented as predecessor activity of processing task since consent always has to be obtained prior to processing. However, if business processes would start to ask for consent from data subject before each processing activity then the information system that uses these processes would be unusable. To resolve this issue, consent is asked once (*e.g.* before user registration or before services are being used for the first time) and for all possible future processing purposes. Hence consent activity could be represented somewhere in the chain flow of previous activities, or in another business process, which logically is being triggered before the actual business process (*e.g.* user registration process - then confirmation of payment process). Similar to recording activity (Chapter 3.3.5) we can support semi-automatic extraction, using property with name 'consent' for each such activity during business process modelling stage. Nevertheless, once consent is extracted from the business process all its Boolean attributes (Figure 1 and Chapter 2.3.4) have to be mapped manually using the context of the process.

***Example**: See Figure 4, processing task 'Create new user' {8} is Processing class, as we already determined. From the context of the sub-process 'Ask for Consent' {5} (Figure 3) we can assume that this is actually an activity to ask for consent, again the 'Ask for Consent' activity is in the flow that is triggered before the processing activity 'Create new user',*

*hence it can be used as consenting activity. Here we assume that consent is given once and for all future processing activities Figure 15.*

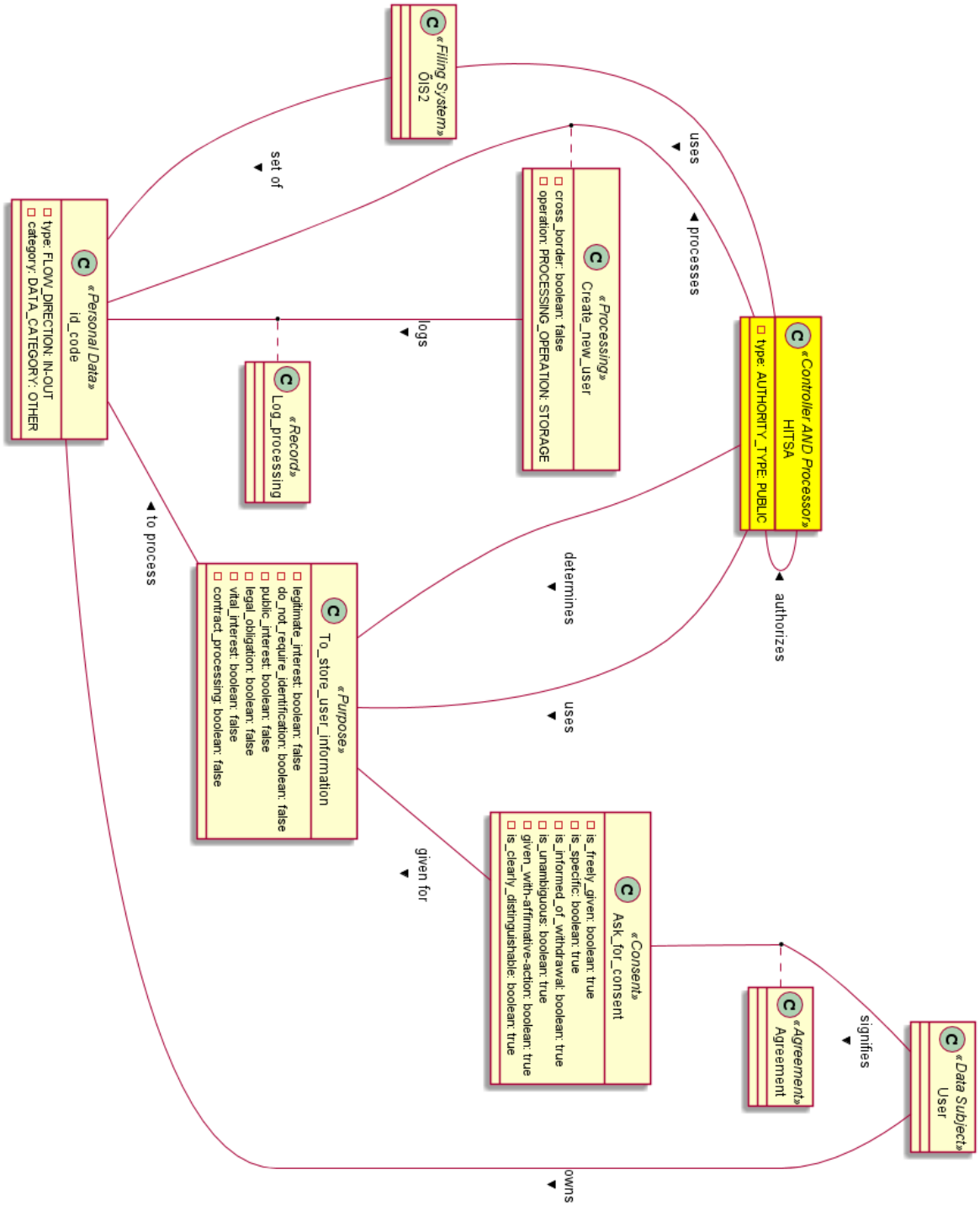


Figure 15. Example ER7

### 3.3.8 ER8. Recipient and Disclosure

Recipient and Disclosure classes are identified as follows. If the personal data object is leaving processing activity in identified Filing System pool via message flow and heading to other pool that is not Data Subject, then receiving pool is considered as Recipient. Disclosure, in this case, is a task from where the data object is leaving. This rule can be implemented automatically.

***Example**: see Figure 4, where activity 'Create new user' {8} has outgoing message flow connections with (i) pool 'External Recipient' {C} and (ii) data object {2} with personal data in it. The result can be seen in* Figure 16.

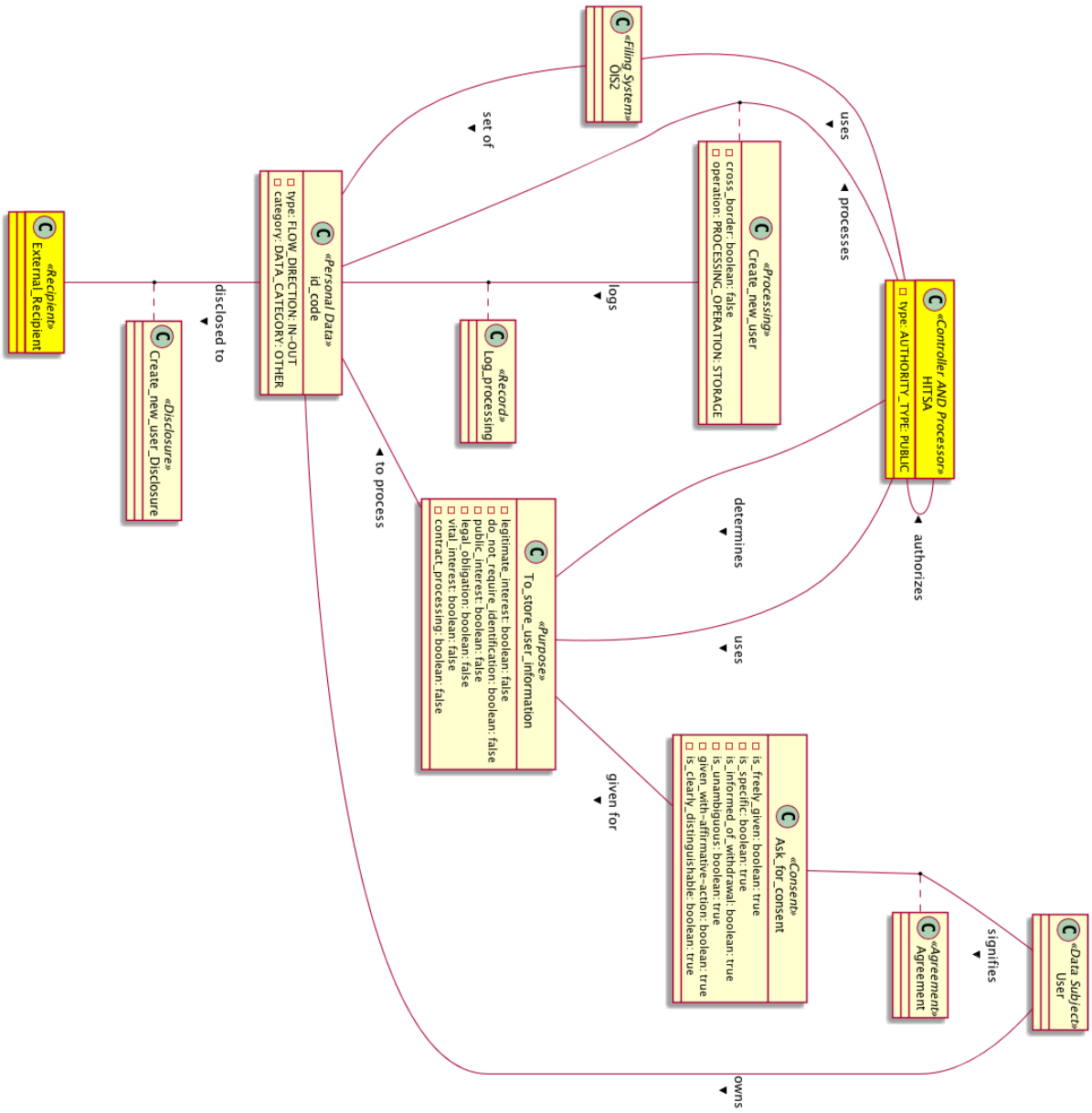


Figure 16. Example ER8

## 3.4 Compliance Validation

This sub-section is focusing on providing details for two steps of the proposed method (Compare two meta-models and Define compliance issues, Figure 2). Once all extraction rules are applied and basis for as-is compliancy model is created, we can resume the compliance analysis and next steps have to be implemented in order to complete validation process: *(i)* application of dynamical constraints (Chapter 2.3.6); *(ii)* evaluation of data subject' rights; *(iii)* comparison of resulting model with meta-model and formulation of possible improvements and issues of business process.

### 3.4.1 Application of Dynamical Constraints

Application of dynamical constraints is very straightforward. Each constraint is formulated in a way that it gives a guidance of how GDPR meta-model has to be modified in order to fulfil validity needs in the certain situation. Constraints can be roughly divided into two groups: *(i)* ones that tighten a final set of validation rules; and *(ii)* ones that loosen a final set of validation rules. For example, **C1** is tightening constraint - it is evaluating whether all Boolean attributes of Consent class are marked as true, if this rule is violated, then as-is compliancy is considered as non-compliant. The example for loosening constraint is **C3**, which evaluates Boolean attributes of Purpose class. If at least one attribute is defined positively then obligatory need for Consent class can be lifted.
Note that during definition of extraction rules we neglected with the declaration of Data Subject' attribute age, which serves as evaluation foundation for constraint **C2**. This was done on purpose. Namely, ignoring attribute age at this stage would keep current as-is compliancy model more abstract, so the model could be used with all Data Subjects, despite their age. Now if we want to concretise as-is compliancy model, then we can use the real age of particular Data Subject and apply constraint **C2**. Otherwise, we would have to repeat the process of information extraction for every Data Subject separately.

### 3.4.2 Data Subject Rights Evaluation

In Chapter 2.3.5 we discussed that data subject's rights can be implemented in information system similarly to personal data processing activities. However, we acknowledge that mechanisms of fulfilling data subject's rights stand apart from data processing activities and most certainly have to be represented as standalone processes. Therefore, we can provide business processes for each right and map these mechanisms to as-is compliancy model. Nonetheless, evaluation of such mechanisms is complicated with the stance of the regulation, where GDPR does not state in any form of whether the mechanism should be certainly integrated into the information system and be available to data subject or can be carried out by usage of different tools and only by processors or controllers. For example, personal data erasure ('right to be forgotten') mechanisms will be implemented differently in social networks and search engines. Users of social networks most certainly will be allowed to remove their accounts and all of the related personal data from the site by themselves, using tools that are provided in the social network. Whereas search engines would not allow such behaviour and there are several reasons for that: *(i)* technical complexity – personal data to be erased can be spanned across several databases and indexes [15] and therefore should be carried out by qualified personnel; *(ii)* web vandalism – allowing users to delete any information form search results, would almost certainly provoke unwanted behaviour of internet trolls, and data that is not considered as personal could be removed as well.
Taking latest discussed arguments in account we cannot provide a universal solution to evaluate mechanisms of implementation data subject's rights. However, in the application of compliance checking in our method, we can acknowledge that these mechanisms exist within observable information system alongside with data processing activities. Figure 17 shows an example business process that can be considered as implementation of Right to Access mechanism in ÕIS2. Firstly, the user makes request {1} to the information system, then ÕIS2 checks whether processing of personal data takes place for this user {2}. If processing takes place, then report {5} should be generated {4} and sent {6} back to the user. The report contains information about purposes of processing, personal data being processed and recipients of related personal data if any. Once the right mechanism is explained it can be mapped to as-is compliancy model, see Figure 18.

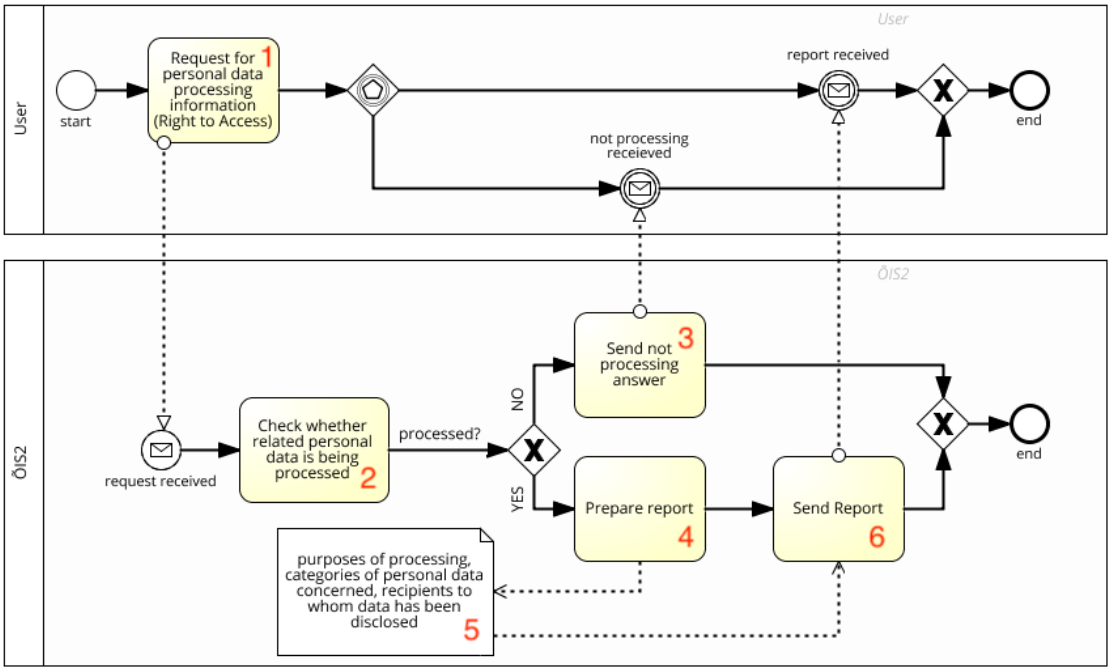


Figure 17. Process for Right to Access

### 3.4.3 Comparison of Two Models and Incompliance Descriptions

Once two models are prepared for comparison, we can now proceed with the validation process. The procedure of comparison is straightforward and follows these three rules (exclusions to rules are defined in sublists):

*(i)* All classes of current meta-model have to appear in as-is compliancy model
*a.* ***Exclusion1****: Abstract classes (Right to, Authority) do not have to appear in as-is compliancy model*
*b.* ***Exclusion2****: Recipient and Disclosure classes are optional and do not have to appear in as-is compliancy model*
*(ii)* All attributes of classes have to be defined
*(iii)* All tightening dynamical constraints have to pass (**C1, C2, C5**)

Based on the absence of classes in as-is compliancy model and failures of the constraints we can highlight different privacy findings. Table 2 list all possible compliance causes and privacy findings for as-is compliancy model.

## 3.5 Summary

In this chapter, we presented an iterative meta-model driven method to check business process compliance to GDPR. The method consists of three compulsory steps and one optional: *(i)* extract as-is compliancy model; *(ii)* compare two models; *(iii)* define compliance issues; and *(iv)* change business process model. We conducted a partial analysis of use case information system being developed and used its business processes to illustrate different steps of the proposed method. With this chapter, we gave an answer to our second sub-question – SUBQ2.

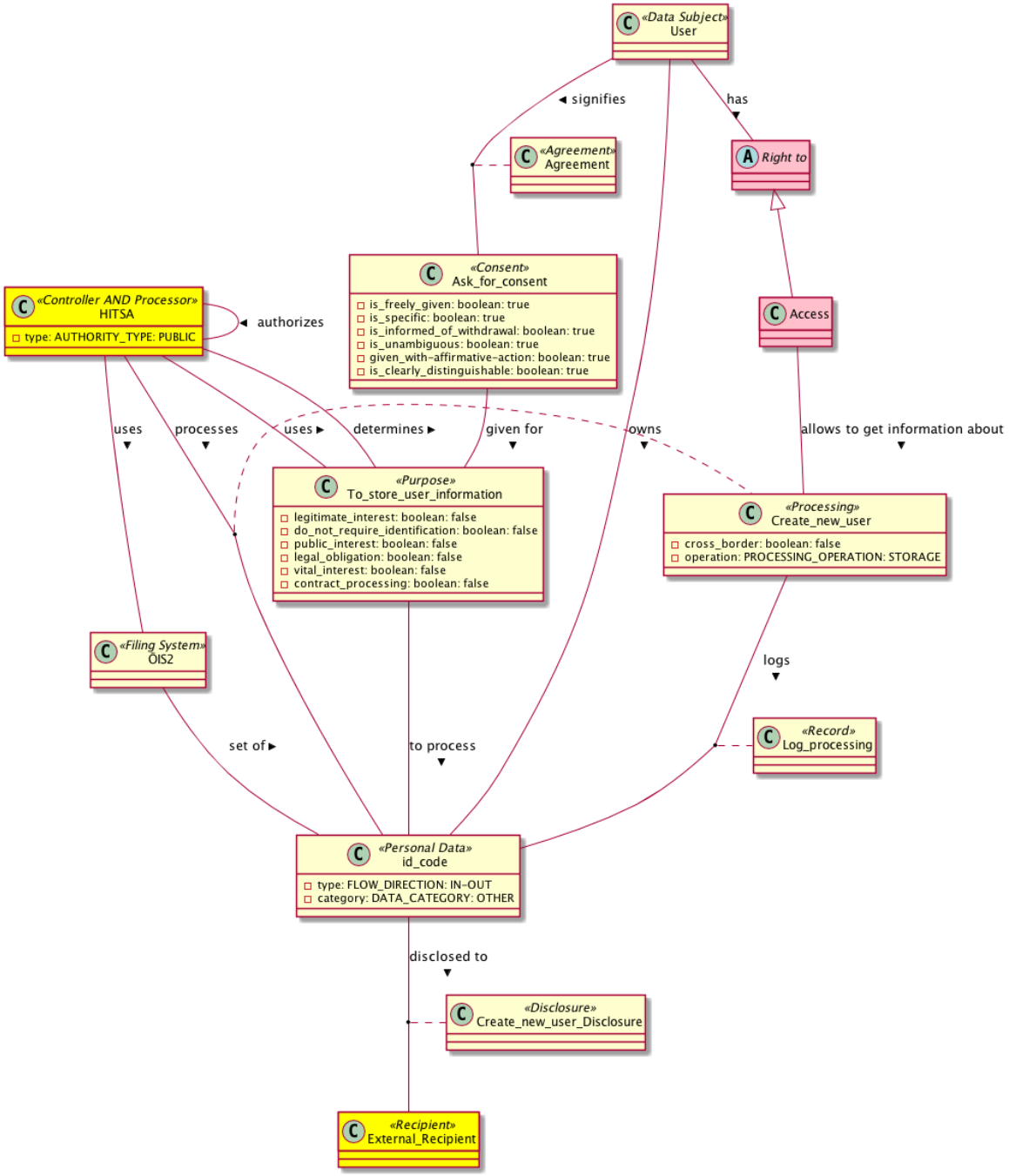


Figure 18. Access Right Added to As-Is Compliancy Model

Table 2. Causes and Incompliance Descriptions

| ID | Cause | Description |
|---|---|---|
| SI1 | Consent class missing | Consent for this processing activity was not collected from the data subject. Data subject should give consent for each purpose of processing his/her personal data prior personal data processing. [Art. 10 GDPR] |
| SI2 | Purpose class missing | The purpose of this processing activity is missing without purpose consent cannot be given by data subject, moreover, processing is prohibited without clear purpose [Art. 4(2), 6(1a) GDPR] |
| SI3 | Record class missing | The processing is not recorded/audited. Each processing activity should be followed by 'recording' activity, which would log processing of personal data. [Art. 30 GDPR] |
| SI4 | Agreement class missing | Agreement for consent was not found. The agreement is needed because processor or controller can be obligated to demonstrate that data subject has consented to his personal data processing. [Art. 7(1) GDPR] |
| SI5 | Data Subject class missing | The data subject is missing. The data subject is a representation of person related to personal data being processed. |
| SI6 | Personal Data class missing | Personal data is missing. The regulation applies only with the processing of personal data. [Art. 2 GDPR] |
| SI7 | Filing System class missing | Filing System is missing. The filing system is a representation of information system, where the business process is being used. |
| SI8 | Processing class missing | Processing is missing. The process does not have any processing activities it is impossible to provide compliance validation. |
| SI9 | Controller class missing | Controller is missing. Controller determines purposes for processing and authorizes processors to process personal data [Art. 4(7) GDPR] |
| SI10 | Processor class missing | The processor is missing. The processor processes personal data on the behalf of the controller. [Art. 4(8) GDPR] |
| SI11 | Rectify class missing | The process to rectify personal data is missing. Provide a process that would allow data subject to rectify inaccurate personal data concerning him/her [Art. 16 GDPR] |
| SI12 | Erase class missing | The process to erase personal data is missing (right to be forgotten). Provide a process that would allow data subject to erase personal data concerning him/her [Art. 17 GDPR] |
| SI13 | Access class missing | Process for data subject to access information about personal data processing is missing. Provide a process that would allow data subject to access information about processing his/her data, including such data as purpose, categories of personal data, recipients etc. [Art. 15 GDPR] |
| SI14 | Restrict Processing class missing | The process to restrict personal data processing is missing. Provide a process that would allow data subject to restrict personal data processing done by this activity [Art. 18 GDPR] |
| SI15 | Portability class missing | The process to export personal data is missing. Provide a process that would allow data subject to access and export personal data being processed/collected by this activity [Art. 20 GDPR] |
| SI16 | Notification class missing | Process to notify data subject about disclosure/rectification/erasure/restriction is missing. Provide a process that would allow data subject to be notified of each rectification, erasure, restriction of personal data processing or disclosure of his/her personal data to third parties or recipients [Art. 19 GDPR] |
| SI17 | Constraint C1 failure | Data Subjects Consent for Processing his/her data was poorly collected [Art. 4(11), 7 GDPR]. |
| SI18 | Constraint C2 failure | Consent cannot be gathered from data subject is younger than 16 years, instead, consent from the holder of parental responsibility should be asked [Art. 8(1) GDPR] |
| SI19 | Constraint C5 failure | This activity is processing personal data relating to criminal convictions and offences, however, authorization to such activity was not granted by official authority. In order to legally process such data, the processor should be authorized by official authority (controller) [Art. 10 GDPR] |

# 4 Prototype Tool for Checking Compliance

The discussed concept and methodology of IS business process compliance checking can be practically proven. Namely, information extraction from BPMN-models and comparison with proposed GDPR meta-model can be automated and implemented using modern web-based technologies. This chapter will show how this can be done and is addressing our third sub-question – SUBQ3.

## 4.1 Design and Requirements

The main concept of the prototype will consist of implementation of the extraction rules and comparison of gathered information with GDPR meta-model. Furthermore, results of validation (issues with compliance, instantiated classes, *etc.*) will be mapped to UML class model and will be shown to the user (see Figure 19).
The tool has to follow the proposed methodology, namely: *(i)* extract information from business process to as-is compliancy model; *(ii)* compare GDPR meta-model to as-is compliancy model; *(iii)* output issues with compliance of business process to the user (see Figure 19).

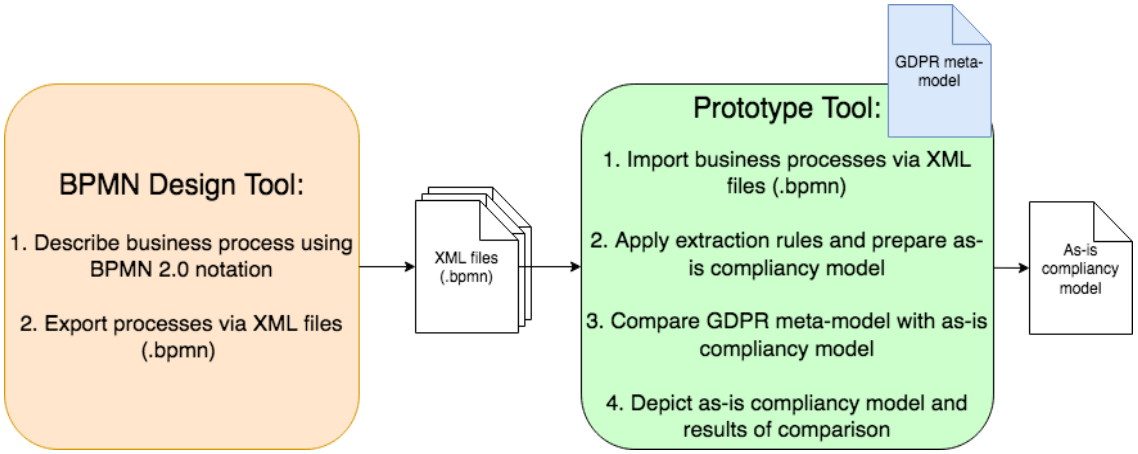


Figure 19. Prototype Tool Scope

To implement extraction rules, the information should be gathered from BPMN business process model inputted by the user. Currently, almost all BPMN modelling tools support unified information interchange file format – XML and standard BPMN 2.0 notation. This means that tool should be able to understand common information interchange format and be able to parse, gather, interpret and transform this information into as-is compliancy model (see Figure 19).

### 4.1.1 User's Perspective and Characteristics

Tool user's perspective can be explained as follows (see Figure 20). The main goal for the user is to have GDPR compliant information system. This can be achieved by using two other software systems (actors): BPMN modelling tool and developed Prototype Tool. The BPMN modelling tool would help to describe business processes and export them using common interchange format, while prototype tool would help to improve business processes by showing compliancy issues of business process to the user.
Since prototype being developed heavily relies on the BPMN and UML notations, tool's users are expected to at least understand both notations and be able to construct and export business processes of the observed IS using some BPMN design tools. Tool users are

expected to be familiar with proposed GDPR meta-model and proposed extraction rules as well.

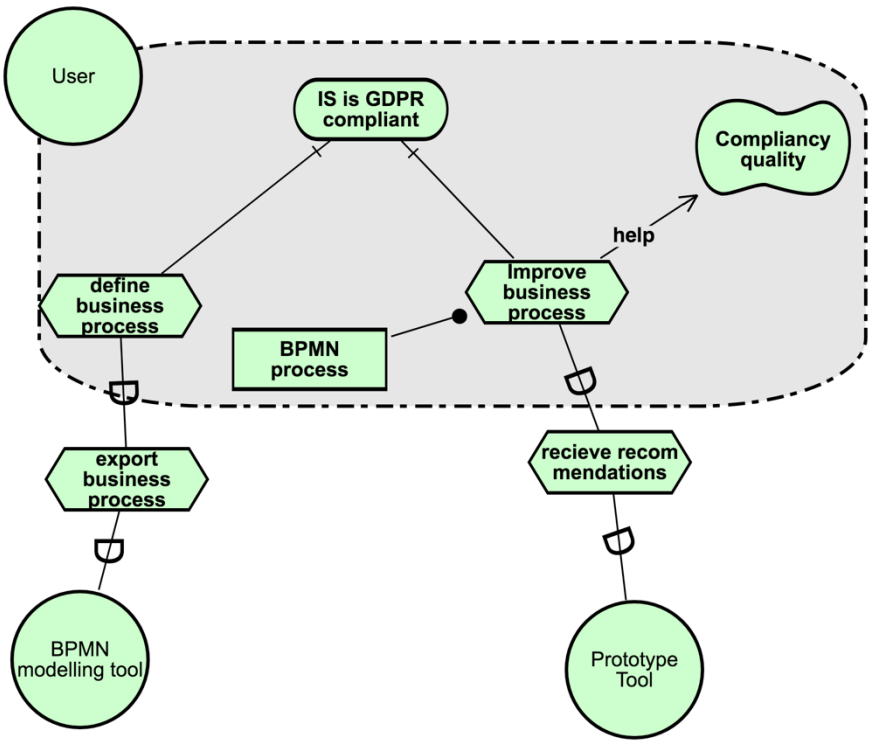


Figure 20. Tool User's Perspective

### 4.1.2 Requirements and Functionality

Based on scope and user perspective these functional requirements will be considered as main core functionality, see Table 3:

Table 3. Functional Requirements

| ID | Description |
|---|---|
| FR1 | The tool should be able to import XML files with BPMN 2.0 notation (XML files with BPMN extension) to prototype tool. |
| FR2 | The tool should be able to parse BPMN 2.0 notation XML files and extract relevant information. |
| FR3 | The tool should be able to analyse several IS at the same time (models). |
| FR4 | The tool should be able to gather additional information from the user (based on extraction rules 3.3):<br>• Information about Controller, Processor and Third Party.<br>• Additional information about Processing and Purpose.<br>• Additional information about Consent.<br>• Information about Filing System if it is needed.<br>• Information about Data Subject's Rights. |
| FR5 | The tool should be able to map extracted information to proposed GDPR meta-model (apply extraction rules 3.3) |
| FR6 | The tool should be able to depict results of compliance validation using UML class diagram notation |

Next use cases will be implemented in prototype tool to meet functional requirements proposed above (see Figure 21):

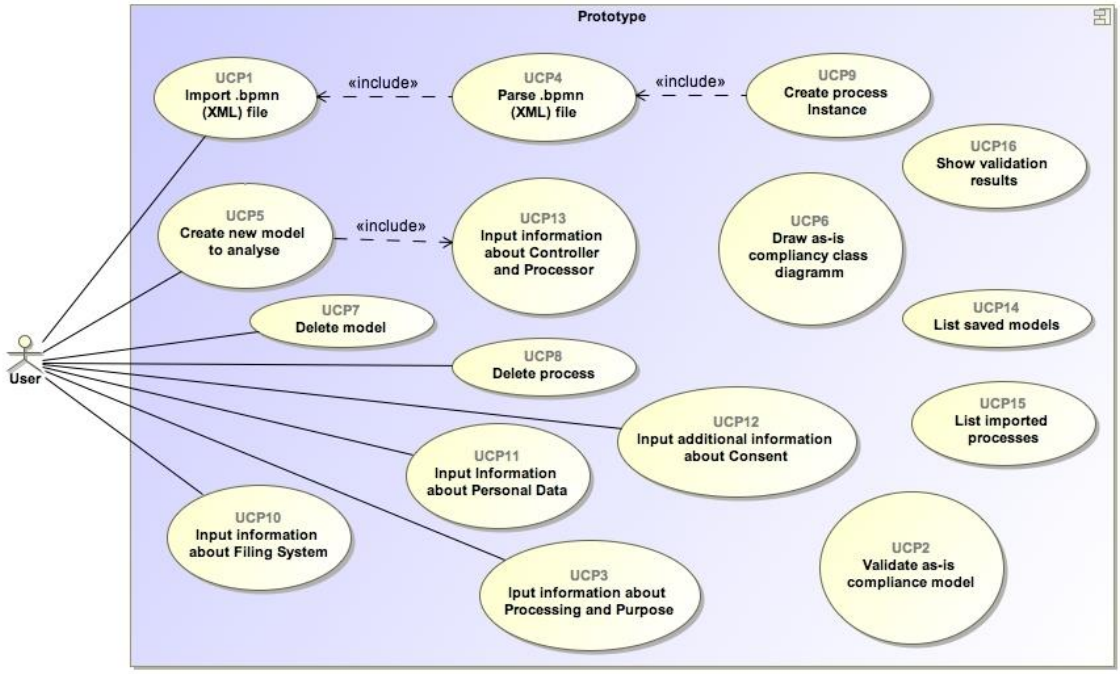


Figure 21. Prototype Use Cases

In Table 4 we present short descriptions of what functionality each use case will cover, however more detailed user stories for each use case can be found in Appendix, Detailed User Stories for Prototype Tool.

Table 4. Use Case Short Descriptions

| ID | Description |
|---|---|
| UCP1 | Functionality will be available for the user and will allow uploading files (with common interchange format XML BPMN) that represent business processes of processing activities or processes that represent data subject's rights. |
| UCP2 | Functionality to validate as-is compliancy model and show inconsistencies of as-is compliancy model compared to meta-model. Moreover, to the user will be shown detailed descriptions for inconsistencies. |
| UCP3 | This use case will provide functionality to collect additional information for class Processing (operation, cross-border) and information for Purpose class (title, Boolean attributes). |
| UCP4 | This functionality will allow the tool to understand what is depicted in the uploaded XML file, that represents a business process. |
| UCP5 | This use case will provide functionality to create a new instance of the model. Model for us is one instance of the observable IS analysis, which can contain several business processes. |
| UCP6 | With this use case will provide functionality to depict as-is compliancy model in UML class diagram notation. |
| UCP7 | Functionality to delete models and all children of the model (processes *etc.*) |
| UCP8 | Functionality to delete process and all children of the process (pools, tasks *etc.*) |
| UCP9 | This functionality will apply the business logic of extraction rules (3.3) to parsed information, which was gathered during UCP4. |
| UCP10 | Functionality to collect information about actors of the business process. Will help to determine whether tool had correctly mapped actors of the business process to as-is compliancy model. |
| UCP11 | Functionality to help gather tool additional information about personal data category. |
| UCP12 | Functionality to collect additional information about Consent class (Boolean attributes) |
| UCP13 | Functionality to collect information about processors, controllers and third-parties. |
| UCP14 | Functionality to list all currently saved models. |
| UCP15 | Functionality to list all currently imported business processes under one model. |
| UCP16 | Functionality to show final results (possible problems) of as-is compliancy model validation to the user. |

### 4.1.3 Architecture

The tool is developed as a web application using classical MVC[6] architectural pattern, as a backbone for this, we used Spring MVC framework[7], which is based on Java programming language. Architectural solution of the prototype can be seen in Figure 22, and tool logics are divided into three main layers:

*(i)* **Data Access** – this layer is responsible for communicating with the database and will be carrying out tasks that are associated with data persistence and data retrieval. The source code is mostly held in two java packages – repository and domain. Domain package will hold classes that are representing objects stored in the database, whereas repository package will hold interfaces that are responsible for data retrieval, persistence and information mapping to domain classes.

*(ii)* **Business** – this layer will serve as a bridge between two other layers – Data Access and View, providing the entire business logic of the program, and will be located under service package. In addition to this, the layer will provide communication with two other external libraries being used in the prototype tool – PlantUML and Camunda.

*(iii)* **View** – this layer will serve as an interface between user and tool through HTTP protocol. The layer logics will be located under web package and will be responsible for rendering web pages and collecting inputs directly from the user.

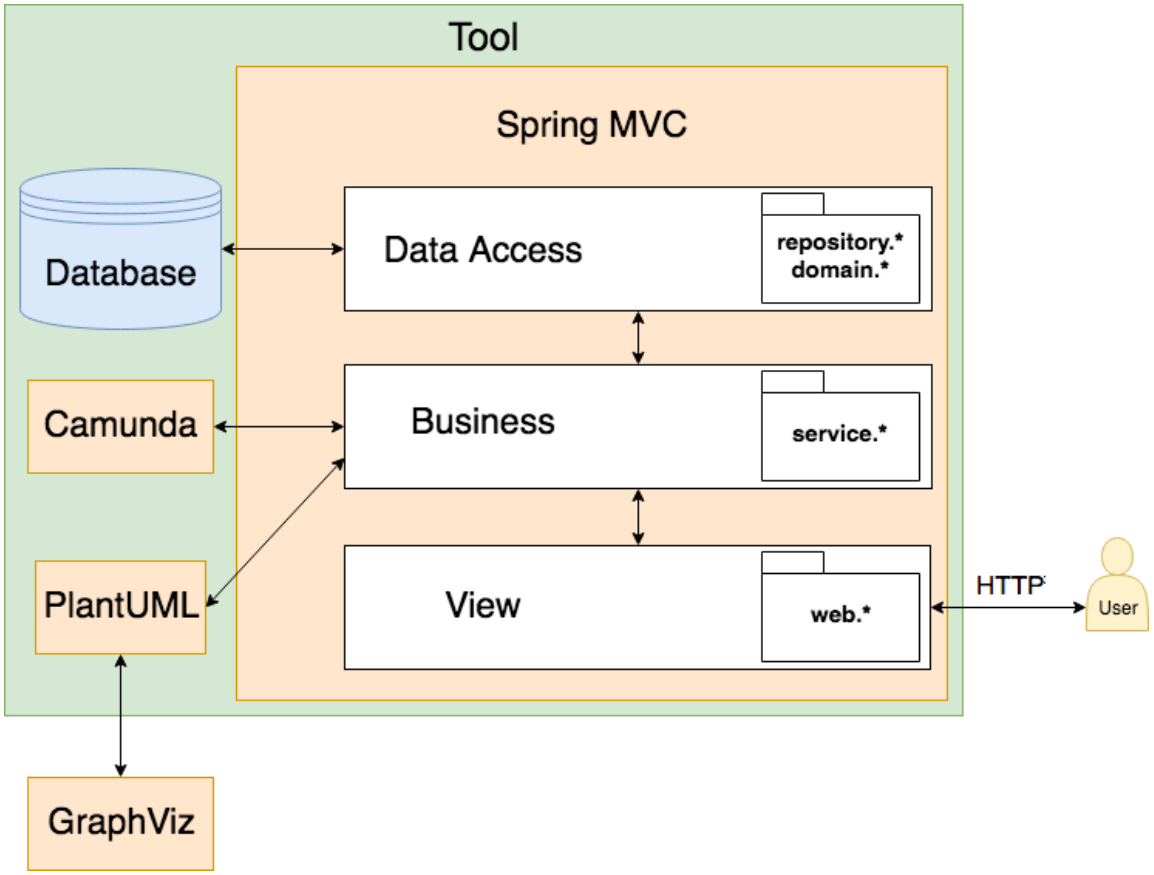


Figure 22. Architecture of Tool

[6] Model-View-Controller is most popular architectural pattern in modern web application development processes

[7] https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html

Other notable parts of the system are:

(i) **Camunda**[8] - external library, that provides functionality to parse XML files with BPMN notation. The tool will use this library to extract information from uploaded business processes.

(ii) **PlantUML**[9] **–** an external library that allows using simple text-based notations to draw UML diagrams, using **GraphViz**[10] visualization software as an external dependency. The tool will use this library to depict as-is compliancy models.

(iii) **Database –** data persistence unit, for developing and testing purposes we used H2[11] in-memory database engine, however, for production purposes, other relational databases can be used.

## 4.2 Implementation

In this sub-chapter, we will give a detailed overview of development decisions made during the implementation process and cover most important functions of source code classes. Source code for prototype tool can be found at[12] and running example sandbox for testing purposes at[13].

Figure 23 shows set of classes that are used in domain package, these classes represent instances of objects that hold information of all models and parsed business processes, short descriptions of classes can be seen in Table 5. Domain package classes are the basis for persistence model and database schema, details for database schema can be found in Appendix, Prototype Implementation. Package domain.enum.* is sub-package of domain package and contains enumeration classes that are used throughout the system as constant values for different purposes, Figure 24.

Table 5. Description for Classes in Domain Package

| Class | Description |
|---|---|
| Model | Representation of one entity of analysis for one IS. |
| BPMNProcess | Representation of business process. Holds information about processing activities and rights of the data subject. |
| ModelAuthority | Intermediate class between Model and Authority to hold information about authority types. |
| Authority | Representation of Authority class in the meta-model. Holds information about controller, processor or third-party entities. |
| Pool | Representation of Pool in BPMN notation. |
| Lane | Representation of Lane in BPMN notation. |
| Task | Representation of Task in BPMN notation. |
| Consent | Representation of Consent class in meta-model. |
| Purpose | Representation of Purpose class in meta-model. |
| TaskPersonalData | Intermediate class to hold direction relations between PersonalData and Task classes. |
| PersonalData | Representation of Personal Data class in meta-model. |
| DataSubject | Representation of Data Subject class in the meta-model. Can hold a direct link to Pool class if the data subject is represented as Pool in the business process. |
| Flow | Representation of flows (message, sequence) in BPMN notation. Holds ids (f – source, t - target) of objects that are connected to each other. |

[8] https://docs.camunda.org/get-started/bpmn20/
[9] http://plantuml.com/
[10] https://www.graphviz.org/
[11] http://www.h2database.com/html/features.html
[12] https://github.com/esgdpr/gdpr
[13] http://138.68.89.5/


Figure 23. Domain Package Classes (domain.*)

Classes that are used in repository package (Figure 25) are interfaces between database and prototype tool, they provide standard methods for domain class persistence and retrieval from the database (findAll, findOne, save *etc.*). However, sometimes standard methods can be overridden to make more complex queries to the database. Examples of overridden queries can be found in interface classes TaskRepository (findAllProcessingTaskByProcessId, findByModelIdAndProcessOrderAndType and findLaneByTaskId), BPMNProcessRepository (getModelIdByProcessId, getFirstByModelAndType *etc.*).

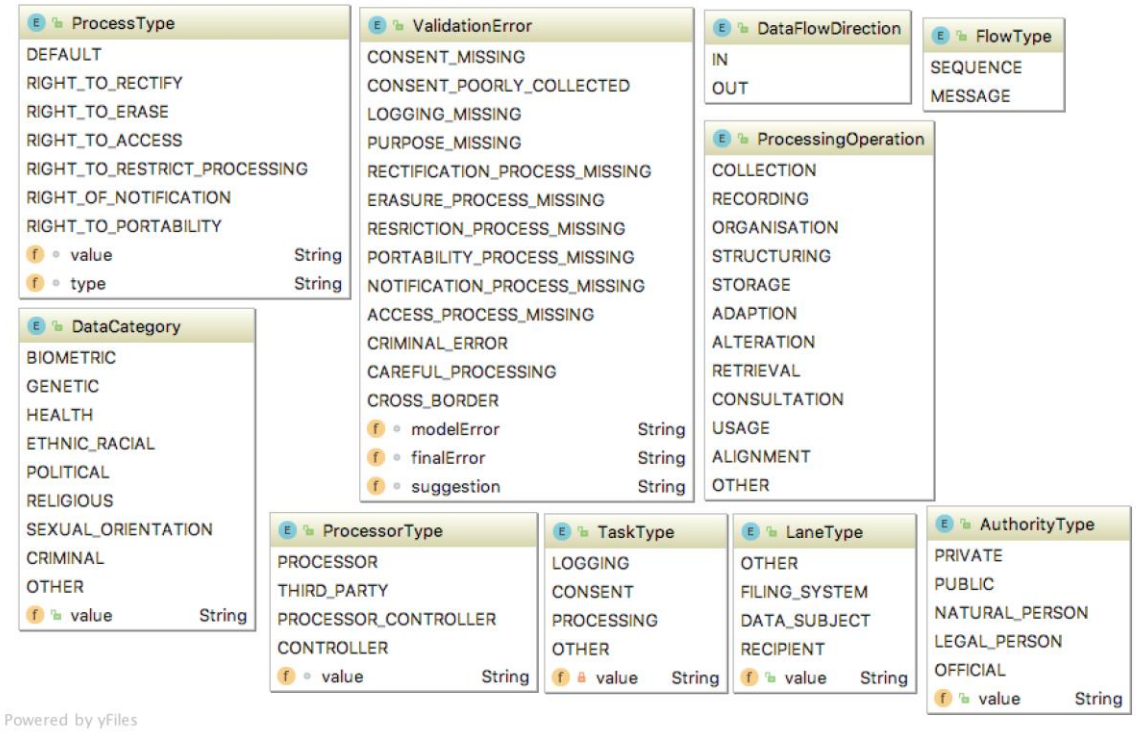


Figure 24. Enum Package Classes (domain.enum.*)

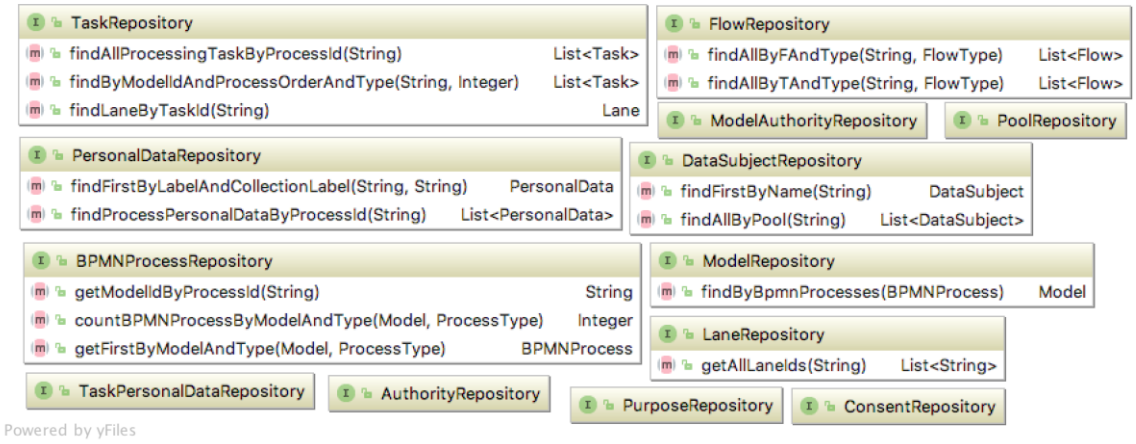


Figure 25. Repository Package Classes (repository.*)

Classes that are used in service package (Figure 26) hold business logic for all functionality within the prototype tool. Detailed descriptions for each class can be found in Table 6.

Table 6. Service Package Classes Description

| Class | Description |
|---|---|
| CommonService | Holds common functions to find different entities. |
| ModelService | Service to handle model creation. |
| ValidatorService | Service to validate as-is compliancy model. |
| ParserService | Holds logics that are used to apply extraction rules and instantiate domain classes. |
| GeneratorService | Holds business logic that generates PlantUML code to depict UML class diagram to represent as-is compliancy model. |
| StepService | Service that holds logic for collecting and saving input information from the user. |
| Note | Intermediate class to represent notes (validation notes) in as-is compliancy model. |

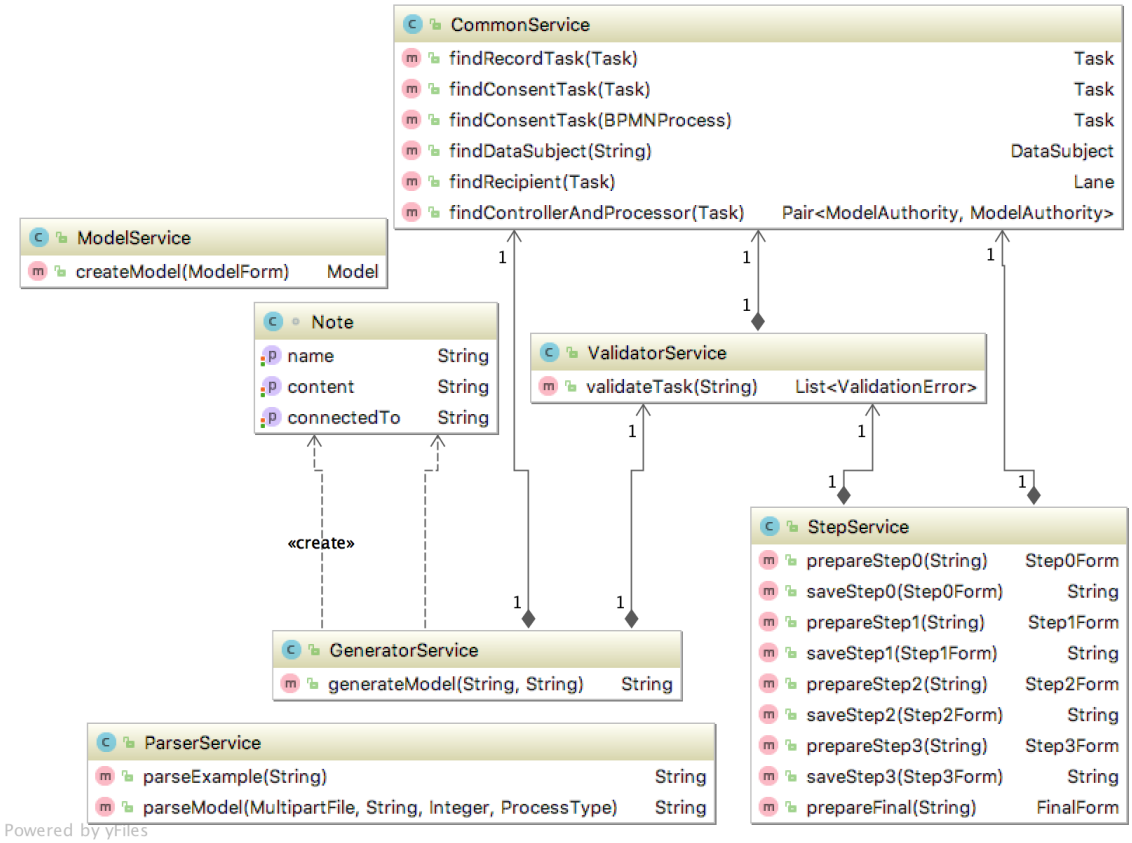


Figure 26. Service Package Classes (service.*)

Classes that are used in web.* package (Figure 27) are responsible for interaction with a user through HTTP protocol, generation of HTML pages and web-friendly representation of different objects. We used Thymeleaf[14] template engine as server-side technology to generate HTML pages and client-side technologies for web-page layout were Bootstrap[15] and jQuery[16].

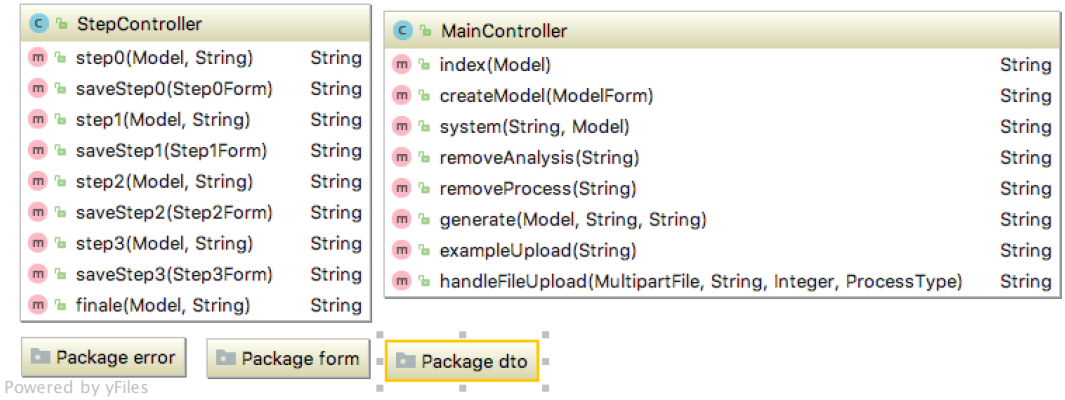


Figure 27. Web Package Classes (web.*)

As we can see, some of the domain classes are resembling BPMN notation objects (Pool, Lane, Task) with minor additions of attributes to be used with meta-model comparison. We

[14] https://www.thymeleaf.org/
[15] https://getbootstrap.com/
[16] http://jquery.com/

decided that it would be a more efficient way to hold data closer to BPMN notation and at the moment of validation map collected information to as-is compliancy model and then conduct validation procedures. More detailed interaction flow of the user, tool and objects can be seen in Figure 28.

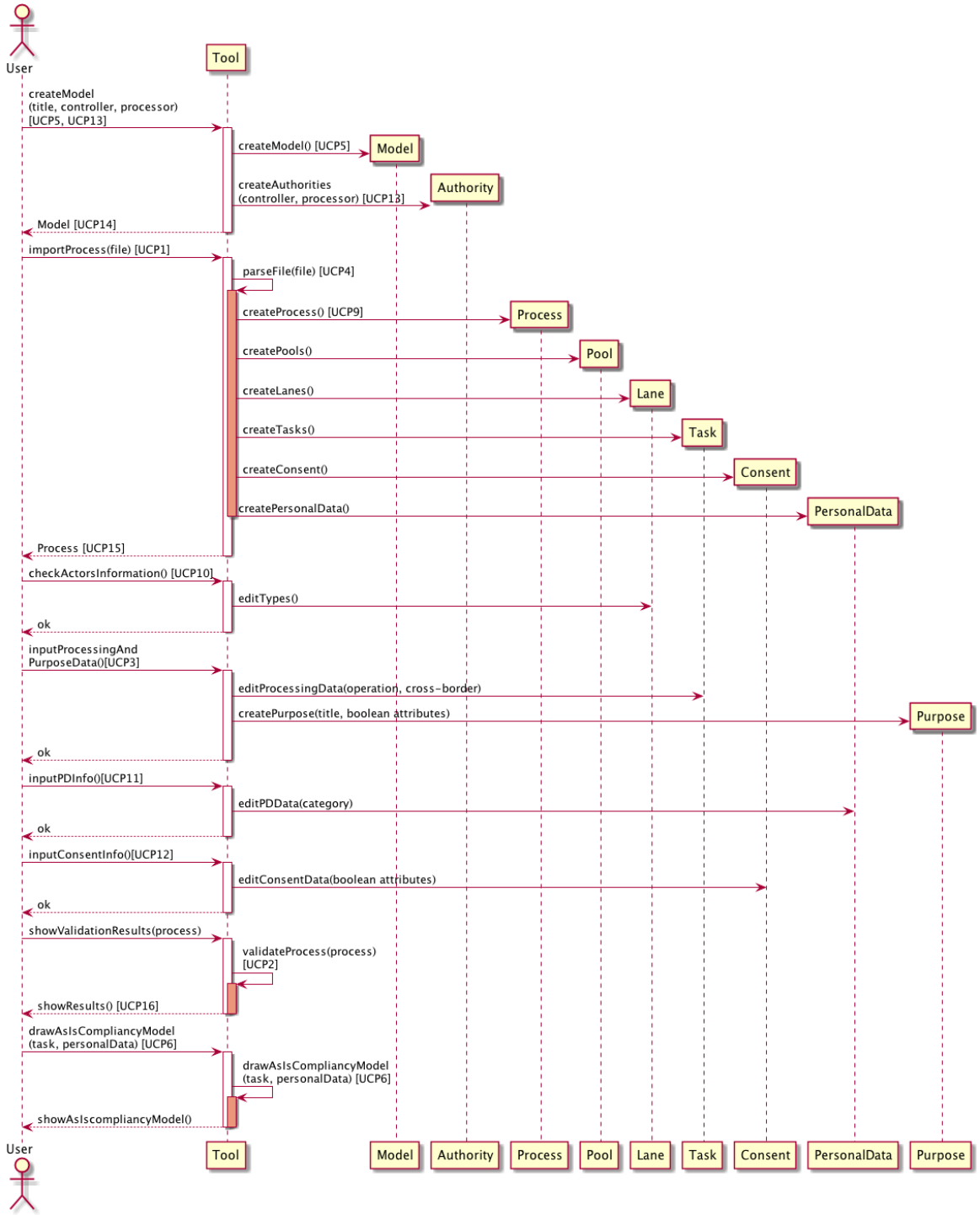


Figure 28. The user, Tool and Objects Interaction Flow

## 4.3 Summary

In this chapter, we presented a way of how proposed method to check business process compliance can be automated. Namely, we developed a prototype tool that is following all steps proposed by the method and is using modern web-based technologies. This prototype tool can be considered as proof of concept to support proposed method to establish process

compliance to GDPR, with this chapter we addressed our third research sub-question SUBQ3. In the next chapter, we will discuss validation possibilities of method and prototype.

# 5 Validation

In this chapter we are focusing on the answer to sub-question SUBQ4, namely, we are performing a validation of proposed method (meta-model-based) to evaluate its performance compared to different methodology. Since our method is shifting towards automated compliance checks it would be interesting to compare its performance against manual methodology - expert opinion-based.

## 5.1 Validation Goals and Process

Validation is conducted to understand and show how the proposed method is performing compared to the expert opinion-based methodology of compliance checking and how usable meta-model-based method is. Based on these goals we are proposing next validation questions, to which we will answer after validation process is completed:

*(i)* Is meta-model-based method of compliance checking underperforming expert opinion-based method?
*(ii)* Is meta-model-based method usable?

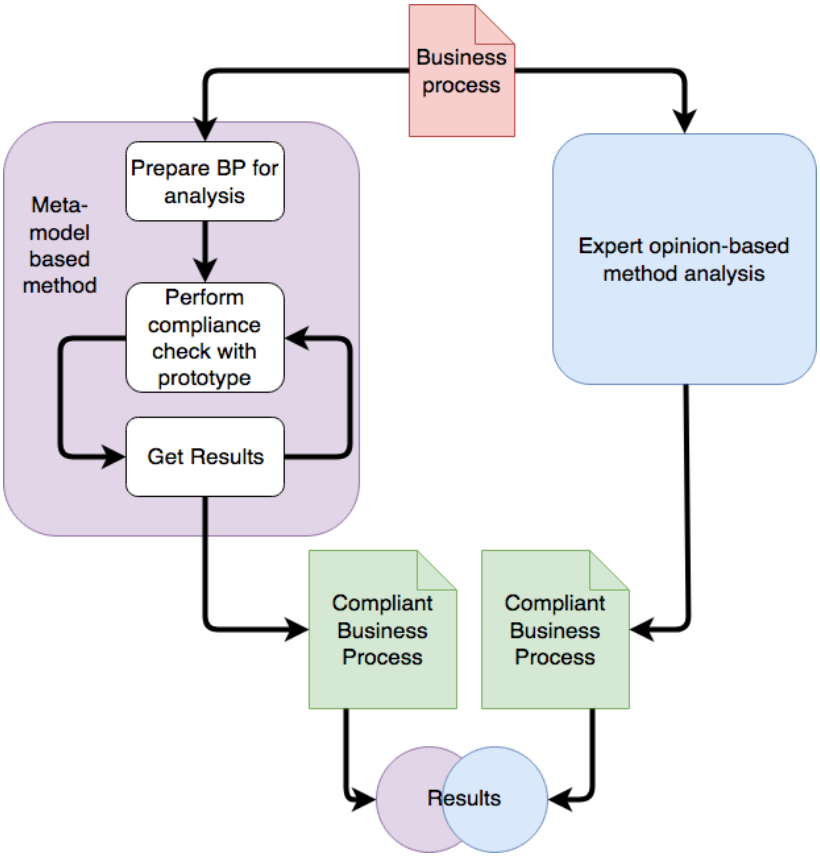


Figure 29. Validation Process

The validation process will be conducted as follows, see Figure 29. Two methods will perform analysis of one non-compliant business process in parallel streams and each method will produce own version of the compliant business process. Once we collect both compliant processes we can compare them with each other, highlight differences and similarities in results and conclude validation. For the sanity of validation two analyses are performed by two different experts, then results will be cross-validated and discussed by both experts until common understanding of validation results is achieved. The meta-model-based method will be conducted by author of this work and expert opinion-based method will be conducted

by A. Abbasi in his Master's thesis work [16]. The meta-model-based method validation (Figure 29) will be conducted using next steps:

*(i)* **Prepare business process for analysis** – some minor modifications have to be made to original process prior analysing it with the help of the prototype tool (Chapter 4).
*(ii)* **Perform compliance check with a prototype** – prototype will be used to check compliance in original business process.
*(iii)* **Get results** – once results are received, modifications to the original business process will be applied. Steps (ii) and (iii) will be repeated until we will be satisfied with the final compliant business process.

## 5.2 Business Process Used for Validation

In our validation process, we will use following business process Figure 34. This business process is currently used by one North-European airline company, where a customer of airline company is making calls to support centre to purchase flight ticket. Process was derived directly from business analysts of the airline company in BPMN notation [16]. Due to GDPR enforcement airline company has been reviewing its business processes and several researches were conducted, including A. Abbasi's research [16], we used this opportunity to cross-validate our method. The process has three main actors:

*(i)* **Customer** – a customer, who wants to purchase flight tickets.
*(ii)* **MBS Agent** – an agent that is working in support centre for an airline company. The agent is serving as an interface between customer and airline information system. From the context given with process, we know that support centre services (MBS company) are outsourced by the airline company and are processing personal data on the behalf on airline company (airline company in this case is controller).
*(iii)* **VDI System** – information system used for searching different flights, registering customers to flights, collecting payment information and generating tickets.

In short business process flow can be described as follows. The customer makes a call to support centre and asks agent for available flights in preferred destinations (A1, A2, A3, and A4). Agent lists all possibilities (B1, B2, B3, B4 and B5) and if the customer is satisfied with proposed results he picks one (A5). Then agent collects personal data from the customer such as name, contact information, date of birth and saves it to airline information system (A7, B7, B8, and B13), provides information of terms and conditions of service (B9). Once this is done customer provides his credit card payment information (number, expiry date, security code) (A8, B10), agent saves this information to VDI system (B14) and issues ticket through same system (B11), then ticket is generated in VDI system (B15) and is sent to customer (B16).

## 5.3 Steps of Meta-Model-Based Method

Our prototype tool is following early proposed extraction rules (Chapter 3.3) of the meta-model-driven methodology and awaits business process to be modelled in a certain way following proposed requirements. The original business process (Figure 35) that is used for validation did not take into account these requirements and some elements (*e.g.* database link with B13 and B14 or data object links to B7 and B8) of the current process have to be remodelled to meet these requirements. In order to conduct validation, we need to slightly modify process (Figure 35). We did not get much context besides what was explained in Chapter 5.2, and we had to make some assumptions to modify process:

*(i)* For us process starts as the offer is made to the customer (A7-B7), and activities that are before that are irrelevant because:

a. All activities before actual offer are done only between MBS Agent and Customer and interaction with VDI System starts from this point.
b. There is no personal data processing involved. At least so states the original business process (Figure 35). Here we are reviewing current business process as one isolated process and not taking into account other business processes that can be performed before, after or simultaneously with the observable process.

*(ii)* We assumed that contact information of customer could mean either a phone number or email, so the ticket could be sent either by email or phone (A7).

*(iii)* We replaced database objects with data objects set notations. We assumed that VDI database would have at least two tables customer and credit_card (B14, B13, and B15).

*(iv)* There is a relation between a credit card and customer in credit card table (see customer_id linked with B14, means that customer can have 0...n credit cards). Since there is a relation between these objects, then it would mean that there is a possibility to identify the customer by credit card information – hence credit card information is personal data as well.

*(v)* We assumed that ticket generation activity (B15) would need at least name and DOB (to print these fields on the ticket) and send ticket activity (B16) would need contact information (to send ticket somewhere).

*(vi)* From the context provided with the business process, we can conclude that Controller, in this case, is Airline, which is a private authority. And the processor is MBS Company which is also private authority (different company that is outsourced by airline company).

*(vii)* Since we are dealing with Airline Company most likely all activities are cross-border activities.

*(viii)* Personal data being processed during this process does not have any special categories (*e.g.* criminal convictions, data concerning health *etc.*)

*(ix)* Since we don't know much details about purposes of processing activities we will consider 'default' case scenario (which expects consent from data subject to process personal data), where each purpose class don't have any of exceptional cases (Boolean attributes, see Chapter 2.3.4).

Validation with prototype tool has highlighted next aspects (Figure 36, Figure 37):

*(i)* Found four processing activities: 'B13 Save personal information', 'B14 Save payment information', 'B15 generate ticket' and 'B16 Send ticket'.

*(ii)* There is no consent asked from the customer prior personal data processing.

*(iii)* There are no logging activities after each data processing activity.

*(iv)* There were no processes uploaded to support rights of the data subject.

To pass validation with prototype tool we managed to do following (Figure 38, Figure 39):

*(i)* Added sub-process (B7a) prior to processing activities, which will be responsible for collecting consent from the customer. Modified process this way that if consent is not given then it ends straight away. However, this activity would work if next assumptions are valid:

a. Since MBS Agent serves as an interface between customer and VDI system then consent could be only asked by MBS Agent. We can assume that consent is given via phone with accepting an answer from the customer, the agreement, in this case, could be the record of the conversation and documents filled and stored in VDI system by MBS Agent.

b. When verbally asking for consent, MBS Agent should give an overview of what personal data of customer will be used and explain purposes for each future processing.

*(ii)* After each processing activity (B13, B14, B15 and B16) we added recording activities (B13a, B14a, B15a and B16a). We assume that VDI system would automatically record each processing count.

*(iii)* We defined processes for each data subject's right. Example of proposed 'Right to Access' method can be seen in Figure 40.

The examples of valid process prototype output can be seen in Figure 41 and Figure 42.

## 5.4 Results of Validation

The expert-opinion based results were following. The compliant process (Figure 43) has implemented a consent activity (G3 and G2) with sub-process (Figure 44). In addition to this, activities to collect payment information from the customer were changed (G4.1 and G4.2 and Figure 45), encryption of personal data (credit card details) was introduced (D3). To the main process were added activities (G5 Say goodbye and document the details, and G6 Send confirmation of data processed and right to be forgotten) these tasks were delegated to MBS Agent and VDI system respectively. Expert-opinion based method proposed activity to restrict personal data processing (Right to restrict processing) as a separate process (Figure 46).

To conclude results of the validation we will give a short overview of similarities and differences in results of compliance checks conducted by two methods. Similarities:

*(i)* Both methods proposed activities to ask consent from the customer, the activity was delegated to MBS Agent. Activities were both arranged that way, so consent would be obtained before actual processing of personal data would take place. If we compare two proposed sub-processes for asking consent (Figure 39 and Figure 44) we would find similarities as well. Namely, the agreement is stored in both cases.

*(ii)* Both methods proposed to implement data subject's rights processes separately as standalone methods (Figure 40 and Figure 46).

Differences:

*(i)* Our method proposed to log processing activities after each processing task (B13a, B14a, B15a and B16a, Figure 38), while other method did not provide such activities.

*(ii)* Other method proposed a different solution for payment and credit card information processing solution with information decryption (G4.1 and G4.2, Figure 43). The rationale for this activity was to skip interaction of MBS Agent with payment information of credit card of customer and provide more secure approach to sensitive data transfer [16].

*(iii)* Prototype insisted on implementation of all procedures that would cover data subject's rights, however, expert-opinion based method covered only consent withdrawal functionality (right to restrict processing). Expert-opinion based method enlists all rights of the data subject to data subject in G6 activity.

Now we can conclude and summarize results of validation by answering questions we defined in Chapter 5.1. Namely:

*(i)* Is meta-model-based method of compliance checking underperforming expert opinion-based method?

With consideration of similarities and differences in two method outputs, we can state that our proposed method does not underperform compared to the expert-opinion based method. In some situations, both methods were very close in implementation of GDPR aspects (Similarities (i, iii)) Meta-model driven method was able to bring out some issues that expert-opinion based method did not cover (Differences (i, iii)) and *vice-versa* expert-opinion based method highlighted some issues that meta-model driven method did not mention (Differences (ii)).

*(ii)* Is meta-model-based method usable?
The fact that we were able to conduct business process validation using prototype tool, which was based on meta-model method, and fact that outputs of different methods had similarities, can state that proposed method is usable. During validation, we discovered some minor software errors and failures in prototype tool. However, discovered software errors were fixed and did not affect validation process.

## 5.5 Threats to Validity

The validation results can be counted as positive, however, threats to validity are still present:

*(i)* We conducted cross-validation with only one business process. To ensure better validation and be sure in final results, more validation processes have to be conducted. Different business processes can provide several new situations and various possibilities to resolve compliance problems, which would help to test our method in variability.

*(ii)* We conducted cross-validation against only one expert. Experts can have diverse points of view to one situation, different knowledge and experience capacity, all this can potentially result in different solutions to one problem.

*(iii)* Meta-model-based method validation part was conducted by its author. This can be considered as a threat because during this work author already had accumulated experience in the modification of business processes to comply with GDPR and knew developed prototype downfalls. For clearer results, validation can be conducted by a person who has smaller experience with GDPR and prototype.

## 5.6 Summary

In this chapter, we discussed validation of a proposed method to establish business process compliance with GDPR through the usage of the prototype tool. The validation goals were to evaluate how our proposed method would perform against expert-opinion based method, and how usable proposed method is. The results of the validation showed that overall performance of the meta-model-based method is comparable to expert-opinion based methodology and we could clearly state that first method does not underperform compared to second. Validation showed that method is usable, however, we cannot state that method can be used by experts that are unfamiliar with GDPR. With this chapter, we addressed our last research sub-question SUBQ4.

# 6 Conclusion

With GDPR enforcement private and state organizations are facing new challenges in the field of personal data processing. One of the greatest challenges is to analyse and adapt business processes to GDPR standards, however, there are no technical solutions proposed to tackle this issue. To address mentioned issue, we formulated our main research question as "How business processes of the information system should be checked for GDPR compliance?" With this work, we gave an answer to main research question by proposing automation friendly method to establish business process compliancy towards GDPR.
The proposed method is based on the meta-model, which we derived from analysis of regulation legislation text. Furthermore, we discussed how discovered meta-model can be used to achieve business process GDPR compliancy. As a result of this, we proposed a meta-model driven method to establish business process compliance. The proposed method is iterative and consists of four steps, where the business process is transformed into as-is compliancy model step by step using extraction rules, and later is compared with meta-model. The results of the comparison can be used to improve the original business process. We supported our findings showing how proposed method can be automated, namely, we developed a prototype tool that follows method's steps and outputs business process compliance analysis results to the user. Furthermore, we conducted cross-validation activity against other methodology and proved that our method, in the face of the prototype, is not underperforming and is actually usable.

## 6.1 Limitations and Lessons Learned

During our work, we encountered several difficulties which resulted in method and prototype limitations. The proposed meta-model is based on the main legislation, and each Member State can expand GDPR to own needs [1], however right now our method does not take it into account. The proposed meta-model is based on our own understanding of regulation legislation text, and since authors are not experienced in jurisprudence, meta-model can be interpreted wrongly. We faced several problems with automation of the method. Namely, we could not provide a universal solution for evaluation of processes that define data subject's rights and all mechanisms still have to be evaluated manually. We could not achieve full automation of our method, however, we proposed ways to partially overcome this issue by introducing several types of extraction rules: automatic, semi-automatic and manual, which is determined by involvement of the user in extraction process. Nonetheless, we still believe that the greater parts of the business process GDPR compliance checks can be automated using our proposed method.

## 6.2 Future Work

This research had limitations, and taking this into account, several offers for future work can be considered:

*(i)* Meta-model expansion and enchantments. GDPR meta-model can be enhanced with cooperative work of lawmakers and software analysts, this way greater parts of misleading legislation text can be resolved and interpreted in a right way. Meta-model can be expanded towards Member States national definitions of GDPR.

*(ii)* Method and Prototype tool improvements. Major improvements of the method can include a definition of additional dynamical constraints and definition of solutions to evaluate data subject rights mechanisms.

*(iii)* Better validation. Validation had several threats to validity and in order to tackle them, better validation could be conducted.

# 7 References


[1] European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council," *Official Journal of the European Union,* vol. 59, no. L119, pp. 1-88, 4 May 2016.

[2] M. F. Gan, H. N. Chua and S. F. Wong, "Personal Data Protection Act Enforcement with PETs Adoption: An Exploratory Study on Employees' Working Process Change," *International Conference on Information Theoretic Security,* vol. 450, no. 1, pp. 193-202, September 2017.

[3] European Union, "Directive 95/46/EC of the European Parliament and of the Council," *Official Journal of the European Union,* no. L281, pp. 31-50, 23 November 1995.

[4] M. Robol, M. Salnitri and P. Giorgini, "Toward GDPR-Compliant Socio-Technical Systems: Modeling Language and Reasoning Framework," in *IFIP Working Conference on The Practice of Enterprise Modeling*, Leuven, 2017.

[5] M.-L. Alaküla and R. Matulevičius, "An Experience Report of Improving Business Process Compliance Using Security Risk-Oriented Patterns," in *The Practice of Enterprise Modeling 8th IFIP WG 8.1. Working Conference*, Valencia, 2015.

[6] V. Diamantopoulou, K. Angelopoulos, M. Pavlidis and H. Mouratidis, "A Metamodel for GDPR-based Privacy Level Agreements," in *Proceedings of the ER Forum 2017 and the ER 2017 Demo Track,* Valencia, 2017.

[7] J. Becker, M. Heddiger, S. Brauer and R. Knackstedt, "Integrating Regulatory Requirements into Information Systems Design and Implementation," in *The International Conference on Information Systems,* Auckland, 2014.

[8] S. Islam, H. Mouratidis and J. Jürjens, "A framework to support alignment of secure software engineering with legal regulations," *Software & Systems Modeling,* vol. 10, no. 3, pp. 369-394, July 2011.

[9] L. T. Ly, F. M. Maggi, M. Montali, S. Rinderle-Ma and W. M. P. van der Aalst, "Compliance monitoring in business processes: Functionalities, application, and tool-support," *Information Systems,* vol. 54, no. 1, pp. 209-234, 2015.

[10] J. García-Galán, L. Pasquale, G. Grispos and B. Nuseibeh, "Towards adaptive compliance," in *SEAMS '16 Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, Texas, 2016.

[11] European Commission, "Proposal for a regulation of the European Parliament and of the Council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation)," 25 January 2012. [Online]. Available: http://ec.europa.eu/justice/data-protection/document/review2012/com_2012_11_en.pdf. [Accessed 29 October 2017].

[12] O. Lynskey, The foundations of EU data protection law, 1, Ed., Oxford: Oxford University Press, 2015.

[13] P. Billgren and L. W. Ekman, *Compliance Challenges with the General Data Protection Regulation,* Lund: Dept. of Informatics, Lund University School of Economics and Management, 2017.

[14] P. Voigt and A. von dem Bussche, The EU General Data Protection Regulation (GDPR): A Practical Guide, Hamburg: Springer, 2017.

[15] D. F. Carr, “How Google Works,” July 2006. [Online]. Available: http://www.baselinemag.com/c/a/Infrastructure/How-Google-Works-1. [Accessed 28 Aprill 2018].

[16] A. Abbasi, “GDPR Implementation in an Airline’s Contact Center,” Master's Thesis, University of Tartu, 2018.

# Appendix

## I. Glossary

| | |
|---|---|
| BPMN | Business Process Model and Notation |
| GDPR | General Data Protection Regulation |
| IS | Information System |
| EU | European Union |
| Member State | The Member State of the European Union |
| Model, Model Instance | A model analysis of IS contains set of process instances |
| PETs | Privacy Enhancing Technologies |
| Prototype, Tool | Software being developed as a proof of concept |
| Process, Process Instance | A single BPMN process entity, parsed and instantiated in a prototype tool |
| PLAs | Privacy Level Agreements |
| UML | Unified Modelling Language |
| XML | Extensible Markup Language |

## II. Running Example Analysis

**Observed business processes.**
From the analysis of ÕIS2, these three business processes are derived to illustrate our method application. These processes are:

1. **User Login**. This process, depicted in Figure 30, is describing the user login into the ÕIS2 system. The user can be logged-in to the system using his ID card or Mobile-ID account (The ID-card and Mobile-ID technology that handles the authentication of users is implemented in various information systems in Estonia, more information of such services can be found at[17]).

   Once the user is authenticated by authentication service (**Login in with mobile ID or ID card,** Figure 30{1}), the system can gain access to his ID-card certificate {2, 3} which contain following information: first and last names, birth date, unique identification code and gender. However, firstly ÕIS2 uses only identification code to check whether the according user is already registered in the system (**Check user ID code,** {4}). If so, then the user can be authenticated in the system and redirected to a personal page (**Authenticate user,** {5}). If identification code was not present in the user database (person table, {6}) then ÕIS2 using information from certificate {8} can create a new record to *person* table {9} (**Create a new user**, {7}) and then authenticate user {5}.
2. **Assign User Permissions.** The process depicted in Figure 31 is describing how a set of permissions can be assigned or removed from ÕIS2 users. Different sets of user's permissions are defining the role of the user in a certain school (*i.e.* student, teacher, administrator *etc.*). These permissions can be added manually by the user with role 'Administrative worker'.

   This process begins with administrative worker login into the system (**User Login** {1}), this process was covered in previous part. After an administrative worker is logged in, he/she has to find the desired person in system database using identification code as input (**Input user ID code** {2} and {3}). The system will run the query (**Check ID code** {4}) against database collection of persons {5} and return response. Now the process can go through two different paths:
   a. **A user with such identification code was not found.** In such cases administrative worker can add a new user manually (**Input user data** {6} and **Save user data** {7}) inputting some relevant user's data, such as name, birth date, email address, gender and phone number {8}. And accordingly, the system will ask for data to be inputted (**Ask to input user data** {9}) and save (**Create user** {10}) this data to appropriate collection {11}.
   b. **A user with such identification code was found.** In this case, the system will show found user's data (**Show user data** {12}, **See user data and permissions** {14, 13}) to an administrative worker using information gathered from several collections, including *person* {11}, *user* and *user_rights* {15}.

   After **a** or **b** path is finished, administrative worker can proceed with adding or modifying user permissions (**Add/modify user permissions** {16}) and saving (**Save user permissions** {17}) them to system, while system will save (**Ask to add/modify permissions** {19}, **Save permissions** {20}) gathered information {18} to collections *user* and *user_rights* {15}.

[17] https://www.sk.ee/en/services/

3. **Application to See Student Data.** In ÕIS2 holder of parental rights of a student is allowed to gain access to see data of student (marks, personal data *etc*.) and process depicted in Figure 32 is describing it.

   In order to gain access to student data, holder of parental rights has to fill application (‘**Input application data’** {1}, ‘**Submit application’** {2}) for access via ÕIS2 attaching relevant information {3}, such as students and parent names, identification codes, means of communication, relationship to student *etc*. Before application is being submitted, ÕIS2 checks whether application is filled correctly (student exists in database, ‘**Check student code ID’** {4}) and whether student data can be accessed by holder of parental rights against information contained in database (student is underage or with special needs {4a}, **Check student age and special needs** {5}). If all these checks are passed, then the application will be stored in the database (**Store application** {6}) and the administrative worker will be notified to take further steps.

   Once administrative worker receives notification of new application he/she will manually **Revise application** {8} and decide whether it is valid or not (based on provided information {3}). If the application is valid then it will be accepted (**Accept Application** {9}) by the administrative worker and later student age will be automatically checked one more time before approval by ÕIS2 (**Check student age** {10}). If checks are passed, then the application will be accepted (**Accept Application** {11}) and access to student’s data will be granted {12}.

   If an administrative worker decides to decline application {13} or ÕIS2 check {10, 14} will fail then justification of decline should be written {15}. Either way holder of parental rights will be notified of decision {16}.

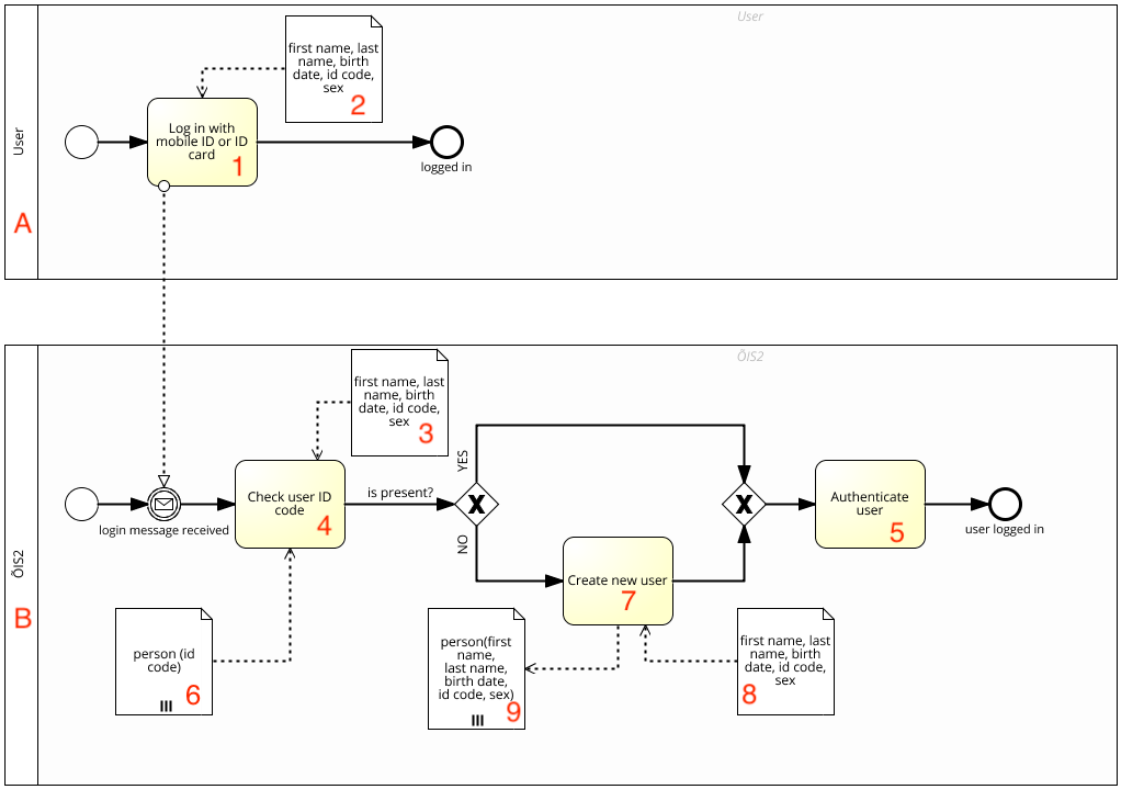


Figure 30.User Login

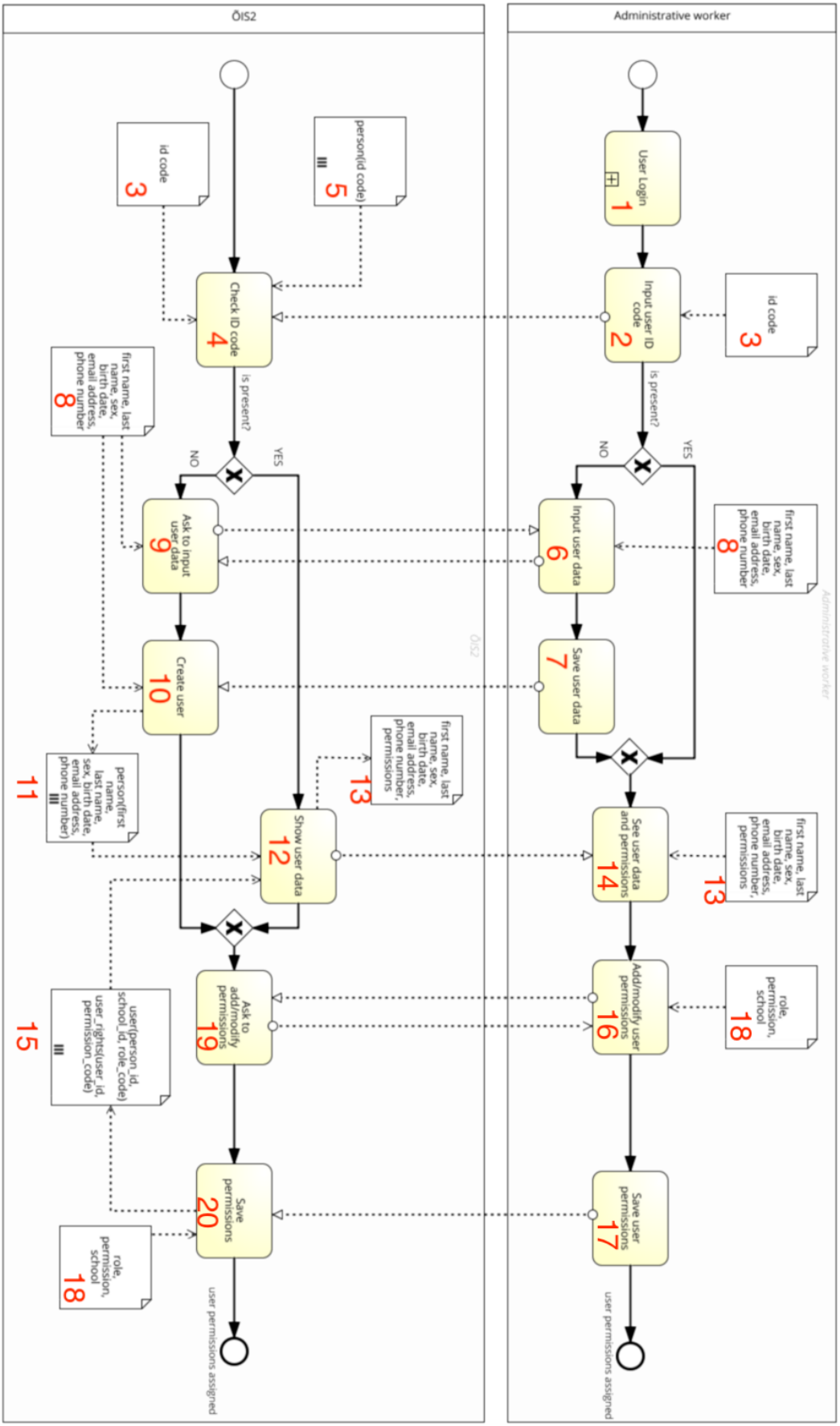


Figure 31. Assign User Permissions

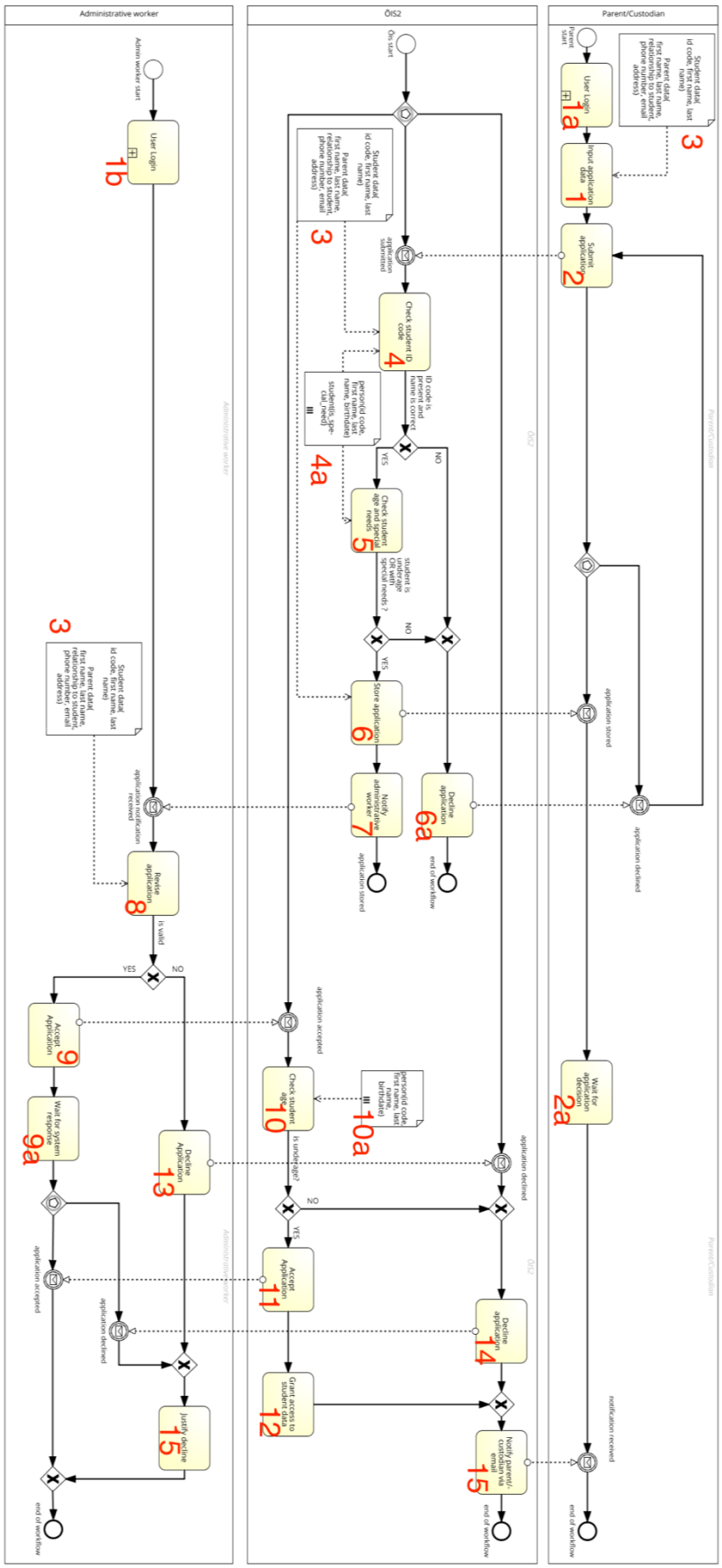


Figure 32. Application to See Student's Data

**Usage of Personal Data.**
Personal data is spanned in several database tables in ÕIS2, however, we will draw our attention to fewer tables since business processes that we are observing are only accessing these tables. Table 7 is representing the storing scheme in ÕIS2 of personal data in observable processes. This overview will help us to understand in what extension personal data can be processed in the observable IS.

Table 7. Tables with Personal Data in ÕIS2

| **Table** | **Field** | **Description** |
|---|---|---|
| person | firstname | First name |
| | lastname | Last name |
| | idcode | The identification code of person (Estonian) |
| | foreign_idcode | The identification code of person (foreign country) |
| | sex_code | Gender of person |
| | citizenship | Citizenship of person |
| | bankaccount | The bank account number of the person |
| | language_code | Studying language |
| | phone | Personal phone number |
| | address | Address of living |
| | residence_country_code | Country of residence |
| | email | Personal email address |
| | native_language | Native language of person |
| | birthdate | Birthdate |
| user_ | person_id | Reference to the person table |
| | school_id | Reference to school table |
| | role_code | The role of the user (*e.g.* Student, parent, admin, teacher) |
| | student_id | Reference to the student table |
| | teacher_id | Reference to teacher table |
| student | person_id | Reference to the person table |
| | school_id | Reference to school table |
| | curriculum_version_id | Reference to curriculum version |
| | study_form_code | Study form code |
| | student_group_id | Reference to study group |
| | language_code | Studying language |
| | is_special_need | Whether a student is with special needs |
| | is_representative_mandatory | Whether a student needs an obligatory representative |
| | special_need_code | Special need code |
| | student_card | Students card number |
| | prevous_study_level_code | Previous study level code |
| | status_code | Status of the student (*e.g.* studying, not studying *etc.*) |
| | ois_file_id | Reference to student photo |
| | curriculum_speciality_id | Reference to curriculum speciality table |
| | study_start | When studying began |
| | study_end | When studying ended |
| | nominal_study_end | Nominal studying ending date |
| | study_load_code | |
| | fin_code | |
| | fin_specific_code | |
| | study_company | |
| | boarding_school | Information about boarding school |
| | student_history_id | Reference to student row modification history |

# III. Detailed User Stories for Prototype Tool

Table 8. Use case UCP1

| | |
|---|---|
| **Name** | Import .bpmn (XML) file |
| **ID** | UCP1 |
| **Description** | This functionality allows the user to import previously defined business processes using BPMN modelling tools into Prototype Tool. |
| **Actors** | User |
| **Triggers** | - |
| **Preconditions** | At least one model is created, and the user is on the model page. |
| **Postconditions** | The XML file is imported, and new process instance is added to this model and shown in the list. |
| **Normal Flow** | 1. User clicks "Browse…" button.<br>2. The user selects a file with BPMN extension from his/her file system.<br>3. The user fills next fields in the form:<br>a. Order of process (number field)<br>b. Type of process (select field)<br>4. User clicks "Submit" button. |
| **Alternative Flows** | - |
| **Business rules** | **BR1**. Fields "file" and "order" are both required. |
| **Exceptions** | **EX1**. The user did not fill one of the required fields:<br>• File - tool shows error "Please select file."<br>• Order - tool shows error "Please enter a number."<br>**EX2.** Tool failed to parse the file:<br>• Error „Something went wrong" is shown to the user |

Table 9. Use case UCP2

| | |
|---|---|
| **Name** | Validate as-is compliance model |
| **ID** | UCP2 |
| **Description** | Tool validates as-is compliancy model comparing it with GDPR meta-model. |
| **Actors** | Tool, User |
| **Triggers** | User goes to "Final" page of the process instance. |
| **Preconditions** | At least one model is created and at least one BPMN process is imported and process instance is created. The user is on the model page. |
| **Postconditions** | Results of the validation are shown to the user. |
| **Normal Flow** | 1. User clicks "Final" button.<br>2. Tool validates as-is compliancy model |
| **Alternative Flows** | - |
| **Business rules** | Business rules of validation are following rules discussed in sub-chapter 3.4 |
| **Exceptions** | - |

Table 10. Use case UCP3

| | |
|---|---|
| **Name** | Input information about Processing and Purpose |
| **ID** | UCP3 |
| **Description** | According to extraction rules, additional information about Processing and Purpose has to be inputted manually, this use case describes this process |
| **Actors** | User |
| **Triggers** | User goes to “Step 1” page of the process instance. |
| **Preconditions** | At least one model is created and at least one BPMN process is imported and process instance is created. The user is on the model page. |
| **Postconditions** | Additional information about Processing and Purposes is saved |
| **Normal Flow** | 1. List of processing task are shown to user<br>2. For every processing task user fills/checks next fields:<br>a. Processing Operation (select field)<br>b. Cross-border processing (checkbox)<br>c. Purpose title (text)<br>d. Legitimate_interest (checkbox)<br>e. Do_not_require_identification (checkbox)<br>f. Public_interest (checkbox)<br>g. Legal_obligation (checkbox)<br>h. Vital_interest (checkbox)<br>i. Contract_processing (checkbox) |
| **Alternative Flows** | - |
| **Business Rules** | **BR2**. Purpose title and processing operation fields are required |
| **Exceptions** | **EX3**. The user did not fill one of the required fields:<br>• Title - tool shows error “Please fill out this field.”<br>• Processing Operation - tool shows error “Please fill out this field.” |

Table 11. Use case UCP4

| | |
|---|---|
| **Name** | Parse .bpmn (XML) file |
| **ID** | UCP4 |
| **Description** | Functionality for a tool to parse inputted BPMN file and instantiate data to process instance (UCP9) |
| **Actors** | Tool |
| **Triggers** | UCP3 |
| **Preconditions** | At least one model is created. BPMN file is inputted for parsing. |
| **Postconditions** | A new process instance is created (UCP9) |
| **Normal Flow** | This functionality is delegated to the external library (Camunda). |
| **Alternative Flows** | - |
| **Business Rules** | - |
| **Exceptions** | - |

Table 12. Use case UCP5

| | |
|---|---|
| **Name** | Create a new model to analyse |
| **ID** | UCP5 |
| **Description** | This functionality to start the analysis of new IS. |
| **Actors** | User |
| **Triggers** | User visits the main page. |
| **Preconditions** | - |
| **Postconditions** | The new model is created and saved to the database. |
| **Normal Flow** | 1. User visits the main page<br>2. User finishes UCP13<br>3. The user is redirected to a created model page (UCP15) |
| **Alternative Flows** | - |
| **Business Rules** | - |
| **Exceptions** | - |

Table 13. Use case UCP6

| | |
|---|---|
| **Name** | Draw as-is compliancy class diagram |
| **ID** | UCP6 |
| **Description** | This functionality allows showing user as-is compliancy model, which was extracted from business process as UML class diagram. |
| **Actors** | Tool, User |
| **Triggers** | User clicks “Generate Model” link at according Processing task and Personal Data instance row (UCP16). |
| **Preconditions** | The user is on the “Final” page of the analysis (UCP16). |
| **Postconditions** | As-is compliancy model is drawn and shown to the user. |
| **Normal Flow** | 1. User clicks “IN” or “OUT” link near each personal data instance<br>2. The tool generates as-is compliancy model using an external library (PlantUML) |
| **Alternative Flows** | - |
| **Business Rules** | **BR3**. Short versions of compliance errors (Table 2) are shown in as-is compliancy model as notes attached to problematic classes, painted in red colour.<br>**BR4**. Generated model code (PlantUML syntax) is shown under the figure of a model |
| **Exceptions** | - |

Table 14. Use case UCP7

| | |
|---|---|
| **Name** | Delete model |
| **ID** | UCP7 |
| **Description** | This functionality will allow the user to delete the model and all its imported processes |
| **Actors** | User |
| **Triggers** | User clicks “Delete” button in the table row, where the model is shown. |
| **Preconditions** | At least one model is created. The user is on the main page. |
| **Postconditions** | Model is deleted from the database. |
| **Normal Flow** | 1. User clicks “Delete” button in the table row, where the model is shown.<br>2. Tool removes the model from the database |
| **Alternative Flows** | - |
| **Business Rules** | **BR5**. All children process of the model is removed as well. |
| **Exceptions** | - |

Table 15. Use case UCP8

| | |
|---|---|
| **Name** | Delete process |
| **ID** | UCP8 |
| **Description** | This functionality allows the user to remove imported process an all children of this process (tasks, personal data *etc.*) |
| **Actors** | User |
| **Triggers** | User clicks “Delete” button in the table row, where the process is shown. |
| **Preconditions** | At least one model is created, and one process is imported. The user is on the model page. |
| **Postconditions** | The process is deleted from the database |
| **Normal Flow** | 1. User clicks “Delete” button in the table row, where the process is shown.<br>2. Tool removes the process from the database. |
| **Alternative Flows** | - |
| **Business Rules** | **BR6**. All children of the process are removed as well (tasks, personal data *etc.*). |
| **Exceptions** | - |

Table 16. Use case UCP9

| | |
|---|---|
| **Name** | Create process instance |
| **ID** | UCP9 |
| **Description** | This functionality is instantiating business process model to classes |
| **Actors** | Tool |
| **Triggers** | UCP1, UCP4 |
| **Preconditions** | - |
| **Postconditions** | Process Instance is created and saved to the database |
| **Normal Flow** | - |
| **Alternative Flows** | - |
| **Business Rules** | **BR7**. Name of the process should be taken from imported filename<br>**BR8**. Only processes with type ‘Personal Data Processing’ are being processed<br>**BR9**. All pools of BPMN processes should be mapped to Pool objects<br>**BR10**. All lane objects of BPMN process should be mapped to Lane objects<br>**BR11**. All task objects of BPMN process should be mapped to Task objects<br>**BR12**. Instantiation of other objects should follow extraction rules. |
| **Exceptions** | - |

Table 17. Use case UCP10

| | |
|---|---|
| **Name** | Input information about Filing System |
| **ID** | UCP10 |
| **Description** | Functionality to help tool to determine actors such as Filing System and Data Subject. |
| **Actors** | User |
| **Triggers** | User clicks “Step 0” at model page (UCP15) |
| **Preconditions** | At least one model is created, and one process is parsed. |
| **Postconditions** | Information is saved. |
| **Normal Flow** | 1. The tool shows form where all detected pools are listed:<br>a. Name - the name of the pool (read-only)<br>b. Type – a type of the pool (select field: FILING_SYSTEM, DATA_SUBJECT, OTHER)<br>2. The user fills all fields and clicks “Save and go to next step” button<br>3. Information is saved, and next step is loaded (UCP3) |
| **Alternative Flows** | **A1**. The user decides not to save progress:<br>2. User clicks one of the following buttons:<br>a. “Back to Models”<br>b. “Back to current model”<br>3. The user is redirected to the desired place. Information is not changed |
| **Business Rules** | - |
| **Exceptions** | - |

Table 18. Use case UCP11

| | |
|---|---|
| **Name** | Input Information about Personal Data |
| **ID** | UCP11 |
| **Description** | Functionality to help tool to determine Personal Data category. |
| **Actors** | User |
| **Triggers** | • User clicks “Step 2” at model page (UCP15)<br>• User clicks “Save and go to next step” at “Step 1” page (UCP12) |
| **Preconditions** | At least one model is created, and one process is parsed. |
| **Postconditions** | Information is saved. |
| **Normal Flow** | 1. The tool shows form where all detected Personal Data instances are listed:<br>a. Label – the label of the Personal data (read-only)<br>b. Category – category of the Personal Data (select field: BIOMETRIC, GENETIC, HEALTH, ETHNIC_RACIAL, POLITICAL, SEXUAL_ORIENTATION, CRIMINAL, OTHER)<br>2. The user fills all fields and clicks “Save and go to next step” button<br>3. Information is saved, and next step is loaded (UCP12) |
| **Alternative Flows** | **A1**. The user decides not to save progress:<br>2. User clicks one of the following buttons:<br>a. “Back to Models”<br>b. “Back to current model”<br>3. The user is redirected to the desired place. Information is not changed. |
| **Business Rules** | - |
| **Exceptions** | - |

Table 19. Use case UCP12

| | |
|---|---|
| **Name** | Input additional information about Consent |
| **ID** | UCP12 |
| **Description** | Functionality to help tool to determine Consent information. |
| **Actors** | User |
| **Triggers** | • User clicks “Step 3” at model page (UCP15)<br>• User clicks “Save and go to next step” at “Step 2” page (UCP11) |
| **Preconditions** | At least one model is created, and one process is parsed. |
| **Postconditions** | Information is saved. |
| **Normal Flow** | 1. The tool shows form with following fields:<br>a. Name – Consent Task name (read-only)<br>b. Consent was freely given (is_freely_given) (checkbox)<br>c. The wording used in agreement for consent was specific and informative (is_specific) (checkbox)<br>d. The wording used in agreement for consent was unambiguous (is_unambiguous) (checkbox)<br>e. Data subject was informed of possibility to withdraw his or her consent at any time (is_informed_of_withdrawal) (checkbox)<br>f. Consent was given with affirmative action (e.g. checking checkbox) (given_with_affirmative_action) (checkbox)<br>g. Request for consent was presented in a clearly distinguishable manner (is_clearly_distinguishable) (checkbox)<br>2. The user fills all fields and clicks “Save and go to next step” button<br>3. Information is saved, and next step is loaded (UCP16) |
| **Alternative Flows** | **A1**. The user decides not to save progress:<br>2. User clicks one of the following buttons:<br>a. “Back to Models”<br>b. “Back to current model”<br>3. The user is redirected to the desired place. Information is not changed.<br>**A2**. There are no Consent instances found in this process or in all sibling processes:<br>1. The tool shows next help text:<br>“Could not find Consent Task in your uploaded .bpmn files. Here is what you can check to fix that:<br>1. If you have Consent Task in one of your .bpmn files then check whether it has attached property with name 'consent'.<br>2. Remember, that order (is determined when .bpmn file is uploaded) of business processes is crucial. The tool searches consent task in previous processes or previous tasks. That means if Consent Task is in the process with order 5, it will be ignored when analysing process with order 4.” |
| **Business Rules** | - |
| **Exceptions** | - |

Table 20. Use case UCP13

| | |
|---|---|
| **Name** | Input information about Controller and Processor |
| **ID** | UCP13 |
| **Description** | This functionality allows gathering information about controller, processor and third parties. |
| **Actors** | User |
| **Triggers** | User visits the main page (UCP14) |
| **Preconditions** | - |
| **Postconditions** | Information is saved. |
| **Normal Flow** | 1. The user is on the main page<br>2. The tool shows form with following fields:<br>a. Controller title (text)<br>b. Controller type (select field)<br>c. Controller fulfils Processor role (checkbox)<br>d. The processor or Third-Party title (text)<br>e. Processor or Third-Party type (select field)<br>f. The processor is Third-Party (checkbox)<br>3. The user fills all fields |
| **Alternative Flows** | - |
| **Business Rules** | **BR12**. If field "Controller fulfils Processor role" is checked then all fields considering processor or third-party should be hidden<br>**BR13**. Controller title, Processor title and types are required fields |
| **Exceptions** | **EX4**. The user did not fill one of the required fields:<br>• The tool shows error "Please fill out this field." |

Table 21. Use case UCP14

| | |
|---|---|
| **Name** | List of saved models |
| **ID** | UCP14 |
| **Description** | This functionality allows the user to see all already created models. |
| **Actors** | Tool, User |
| **Triggers** | The user has visited the main page |
| **Preconditions** | - |
| **Postconditions** | List of saved models is shown to the user. |
| **Normal Flow** | 1. User visits the main page<br>2. List of all previously created models is shown to the user as a table with next columns:<br>a. # - sequential number<br>b. Model – the title of the model<br>c. Options – options to manipulate models:<br>i. Open – opens selected model<br>ii. Delete – deletes selected model (UCP7) |
| **Alternative Flows** | **A1**. There are no previously created models in database:<br>2. The message "There are no models created yet. You can create a new model with the form above (Create a new model to analyse)." is shown to the user |
| **Business Rules** | - |
| **Exceptions** | - |

Table 22. Use case UCP15

| | |
|---|---|
| **Name** | List imported processes |
| **ID** | UCP15 |
| **Description** | This functionality allows the user to see all processes uploaded under model |
| **Actors** | Tool, User |
| **Triggers** | User goes to a model page |
| **Preconditions** | At least one model is created, and one process is parsed. |
| **Postconditions** | List of all processes listed in the model is shown. |
| **Normal Flow** | 1. User goes to model page clicking "Open" at table row, where models are listed.<br>2. All processes that belong to the model are listened to a table with next columns:<br>a. # - sequential number<br>b. Filename/Process name – the name of file or process<br>c. Order – order of the process<br>d. Options – options to manipulate the process<br>i. Step 0 – Go to UCP10<br>ii. Step 1 – Go to UCP3<br>iii. Step 2 – Go to UCP11<br>iv. Step 3 – Go to UCP12<br>v. Final – Go to UCP16<br>vi. Delete – UCP8 |
| **Alternative Flows** | **A1**. There are no previously imported processes in database:<br>2. The message "There are no business processes uploaded yet. You can upload business processes using the form above (Upload new .bpmn model here)." is shown to the user |
| **Business Rules** | - |
| **Exceptions** | - |

Table 23. Use case UCP16

| | |
|---|---|
| **Name** | Show validation results |
| **ID** | UCP16 |
| **Description** | This functionality allows showing user results of the validation (descriptions of incompliance). |
| **Actors** | User, Tool |
| **Triggers** | User visits "Final" page |
| **Preconditions** | At least one model is created, and one process is parsed. One data processing task is found in the business process. |
| **Postconditions** | Validation results are shown for each processing class. |
| **Normal Flow** | 1. User clicks "Final" button on the model page (UCP15)<br>2. Tool validates (UCP2) as-is compliancy model and shows results to the user |
| **Alternative Flows** | - |
| **Business Rules** | **BR14**. Incompliance descriptions for validations are taken from Table 2 |
| **Exceptions** | - |

# IV. Prototype Implementation

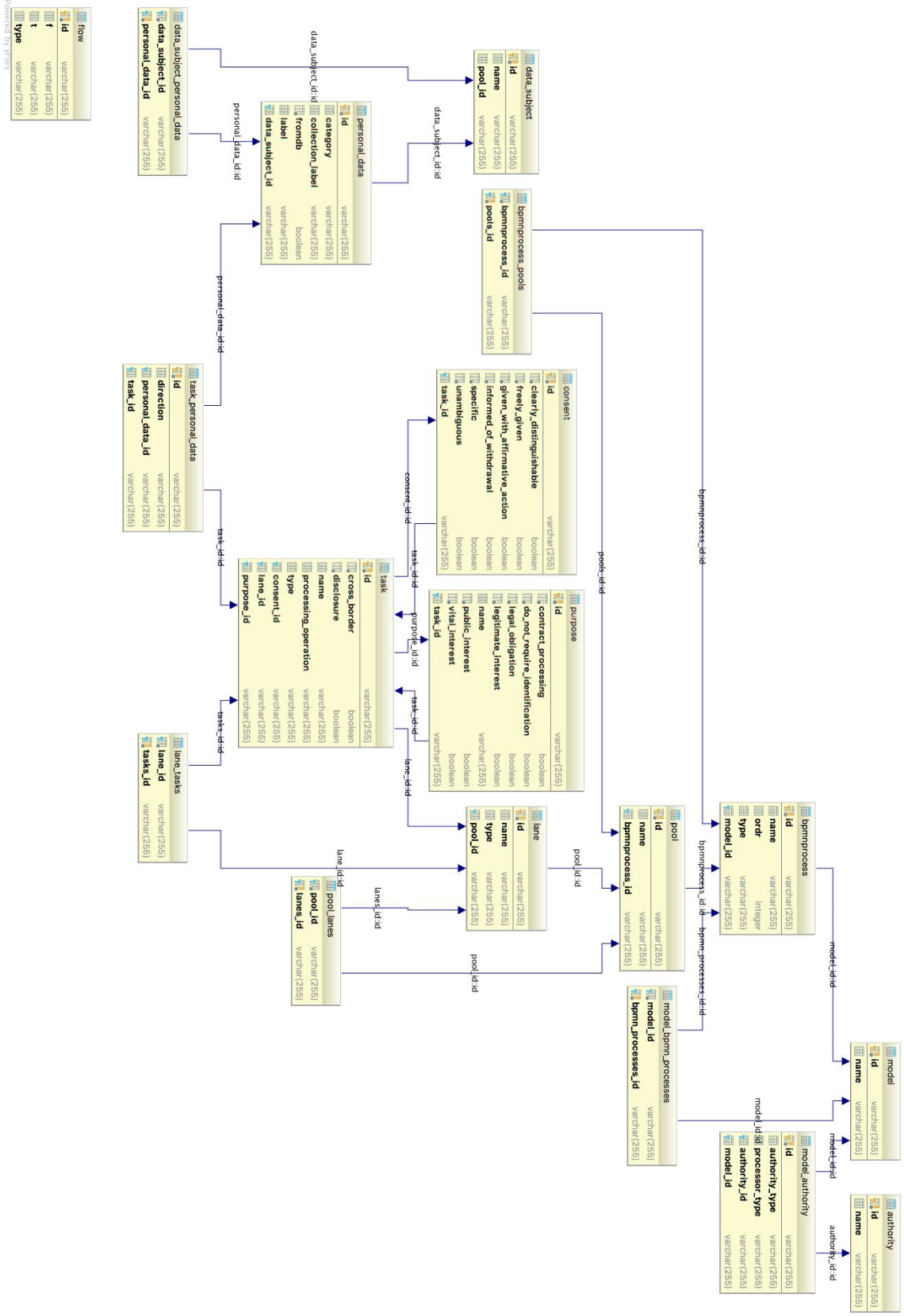

Figure 33. Database Schema

# V. Validation Additions

Figure 34. Process Used in Validation

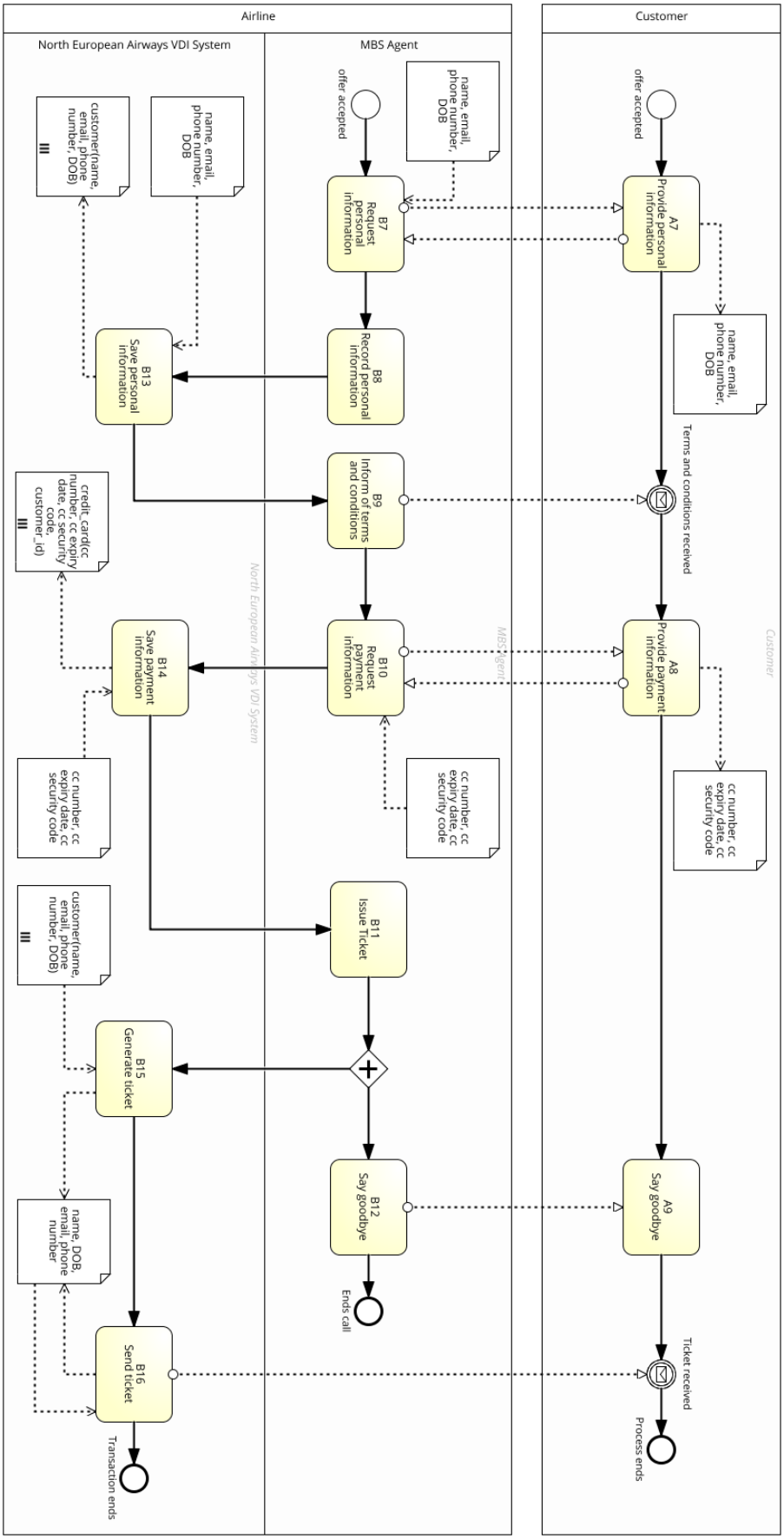


Figure 35. Modified Process to Use with Prototype


Figure 36. Example Output from Prototype Tool

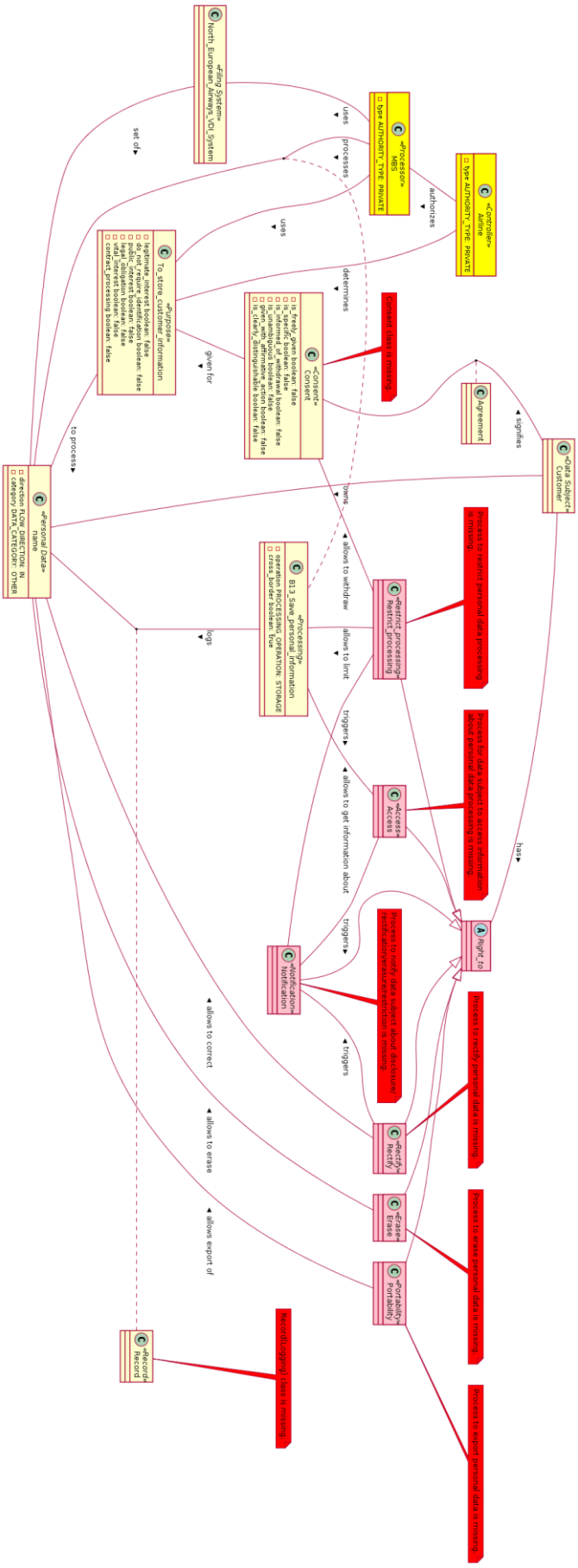


Figure 37. Example of Generated As-Is Compliancy Model

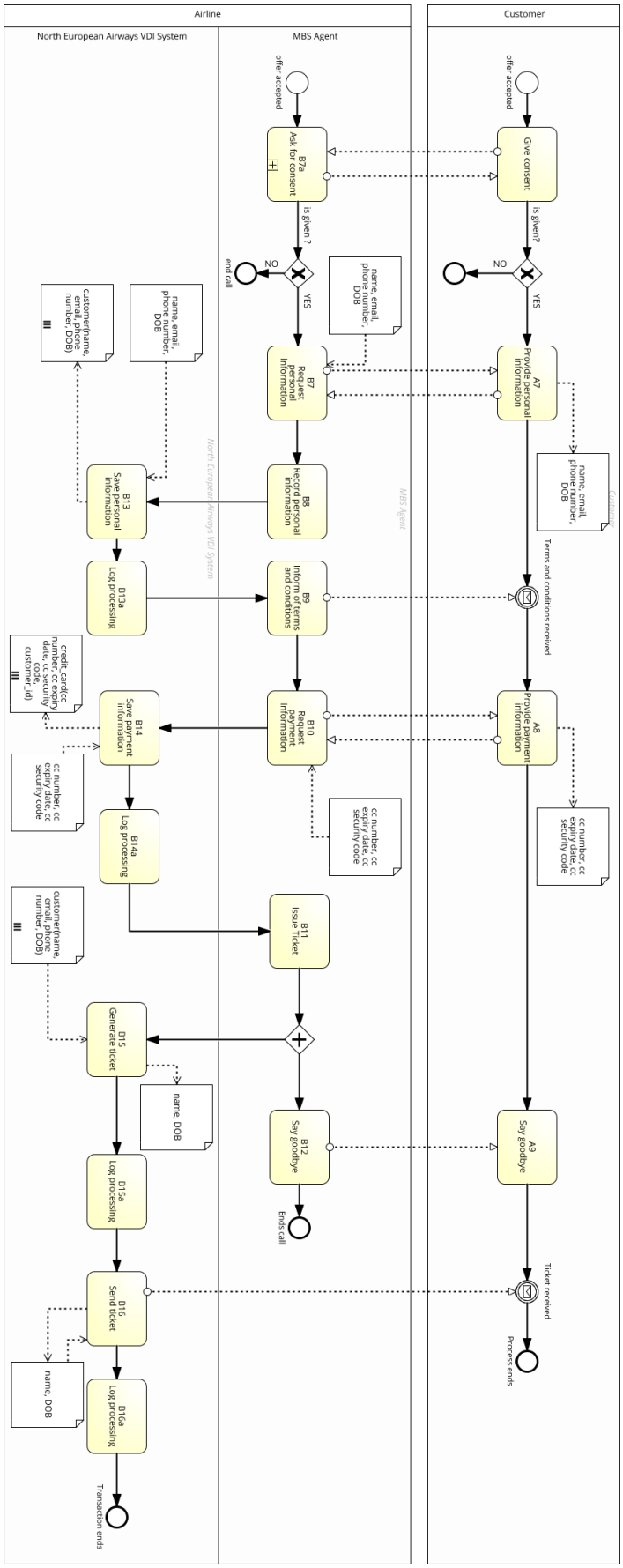


Figure 38. Modified Process to Pass Prototype Validation

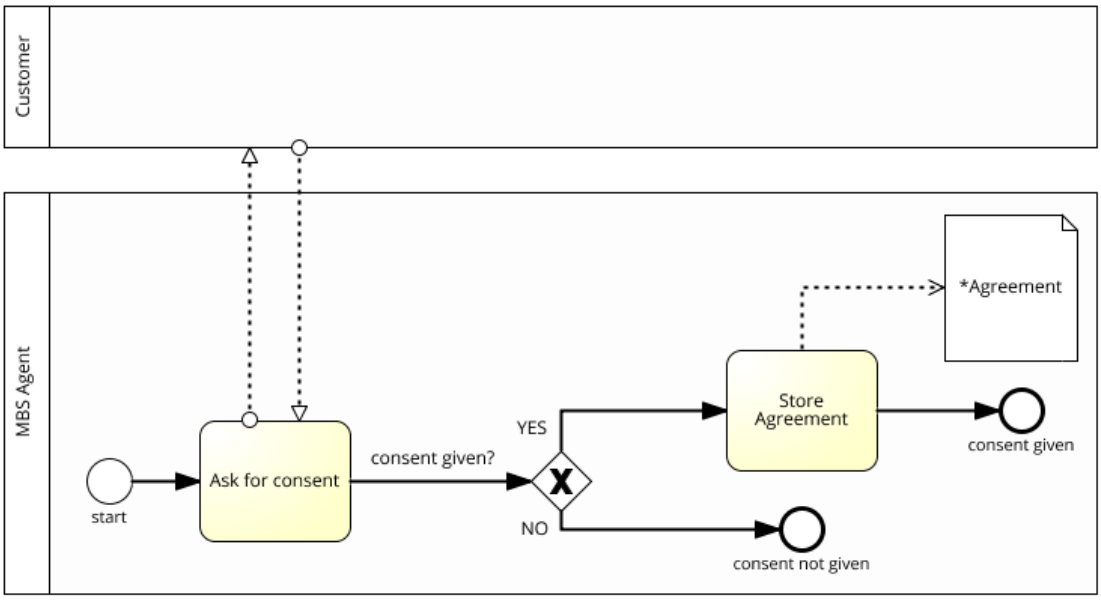


Figure 39. B7a Ask for Consent Sub-process

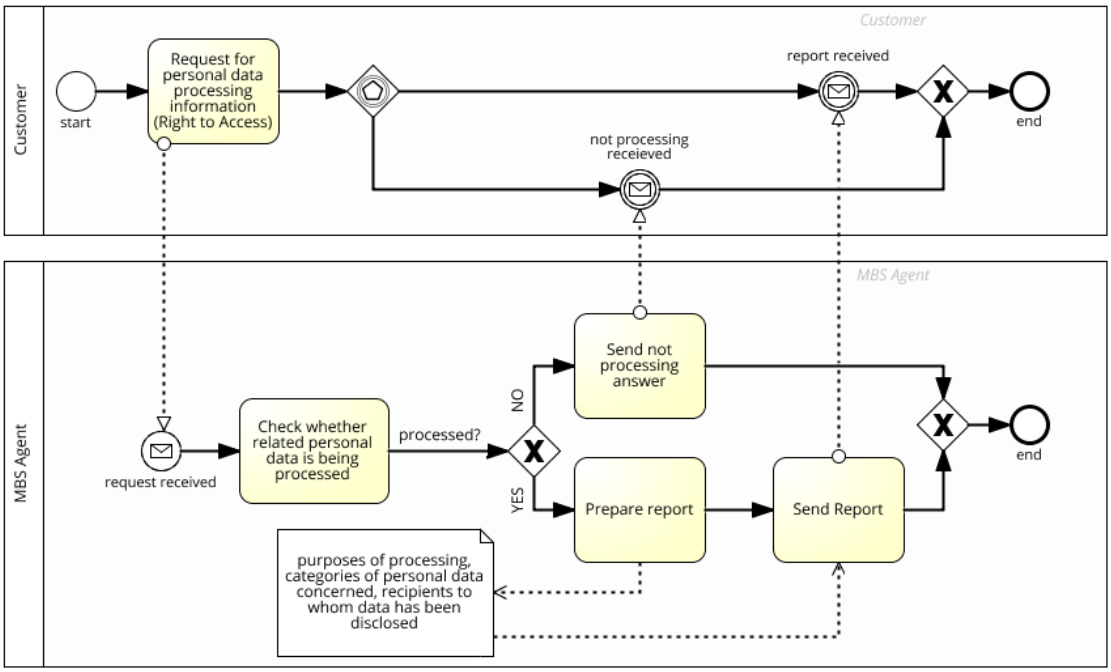


Figure 40. Example of Access Method

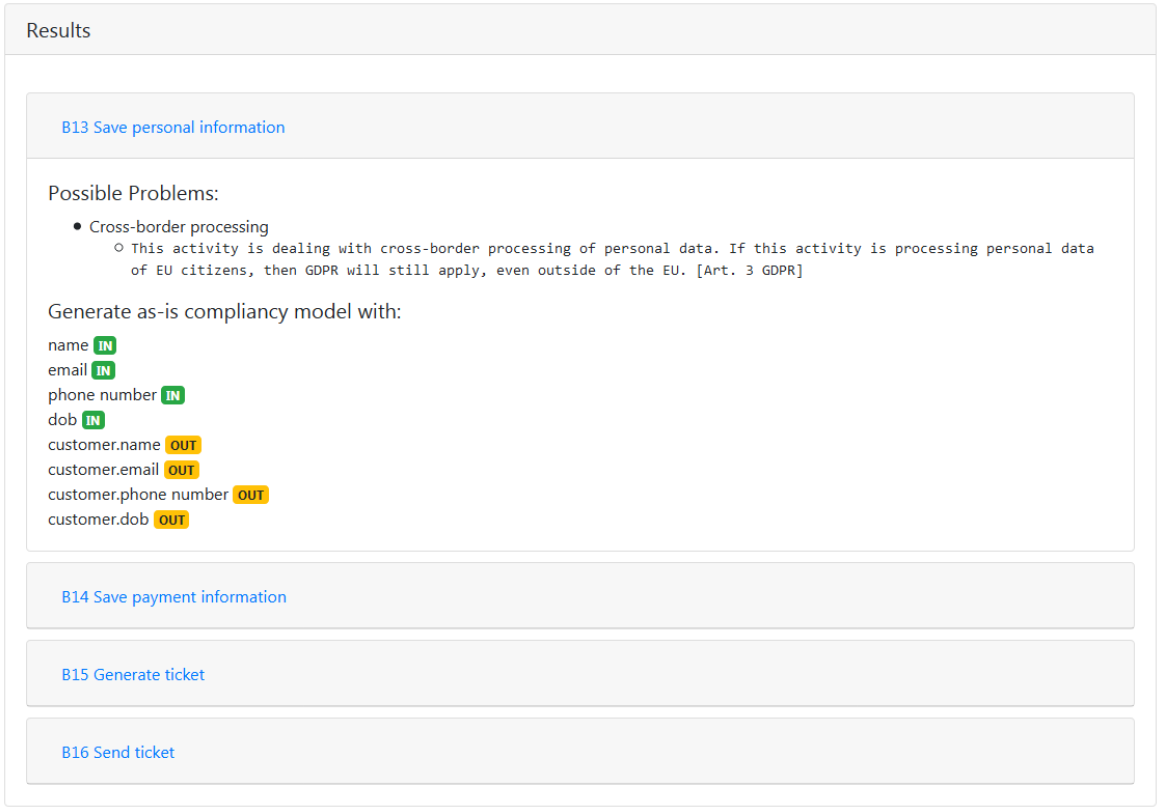


Figure 41. Prototype Output after Process Was Changed

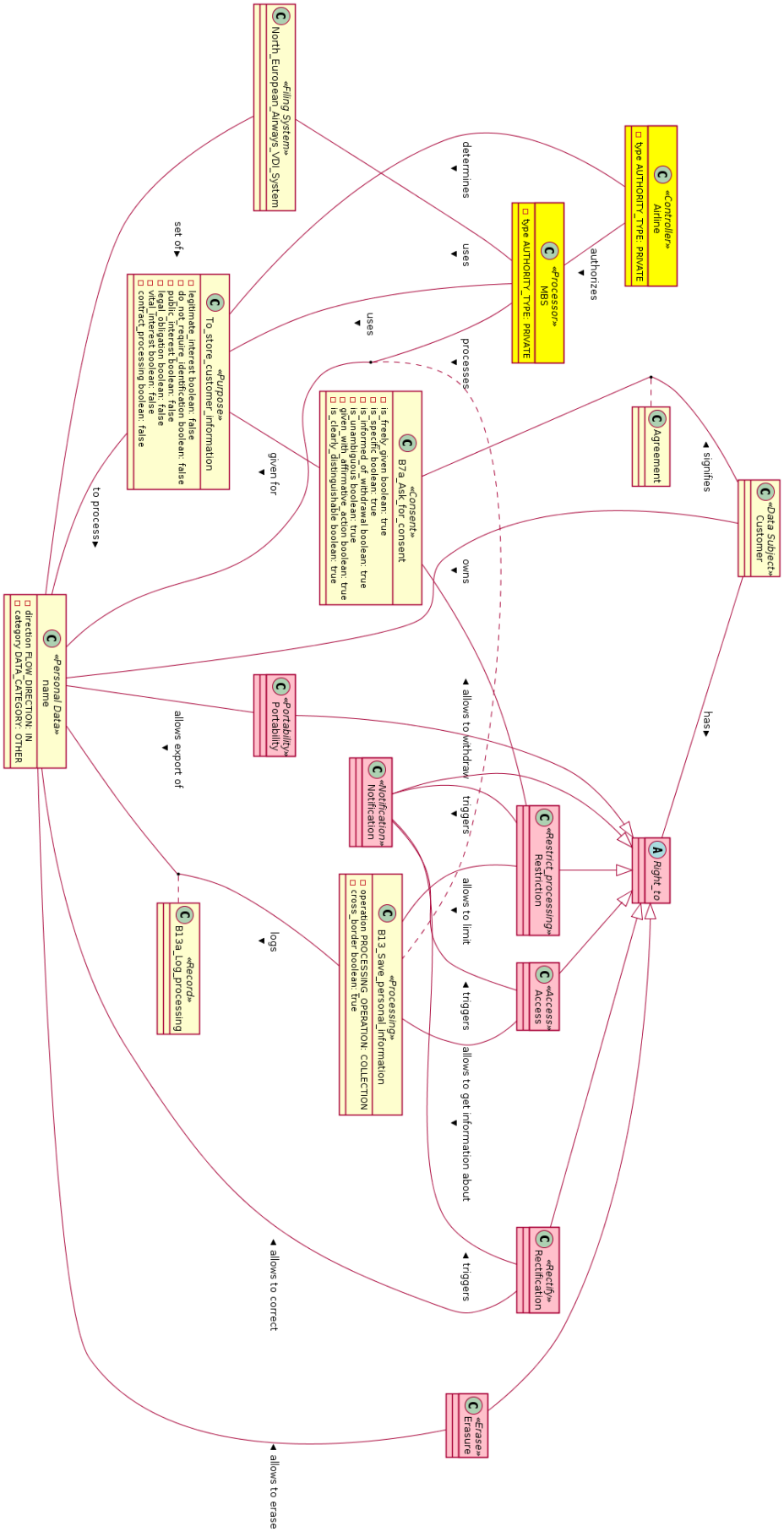


Figure 42. As-Is Compliancy Model After Process Changed

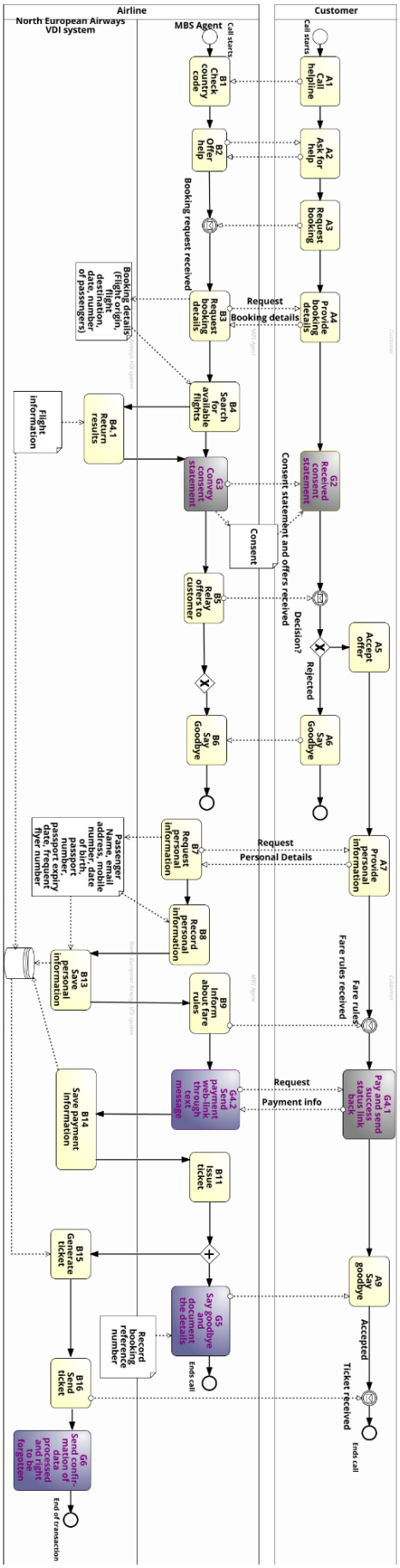


Figure 43. Compliant Process Expert-Opinion Based Method

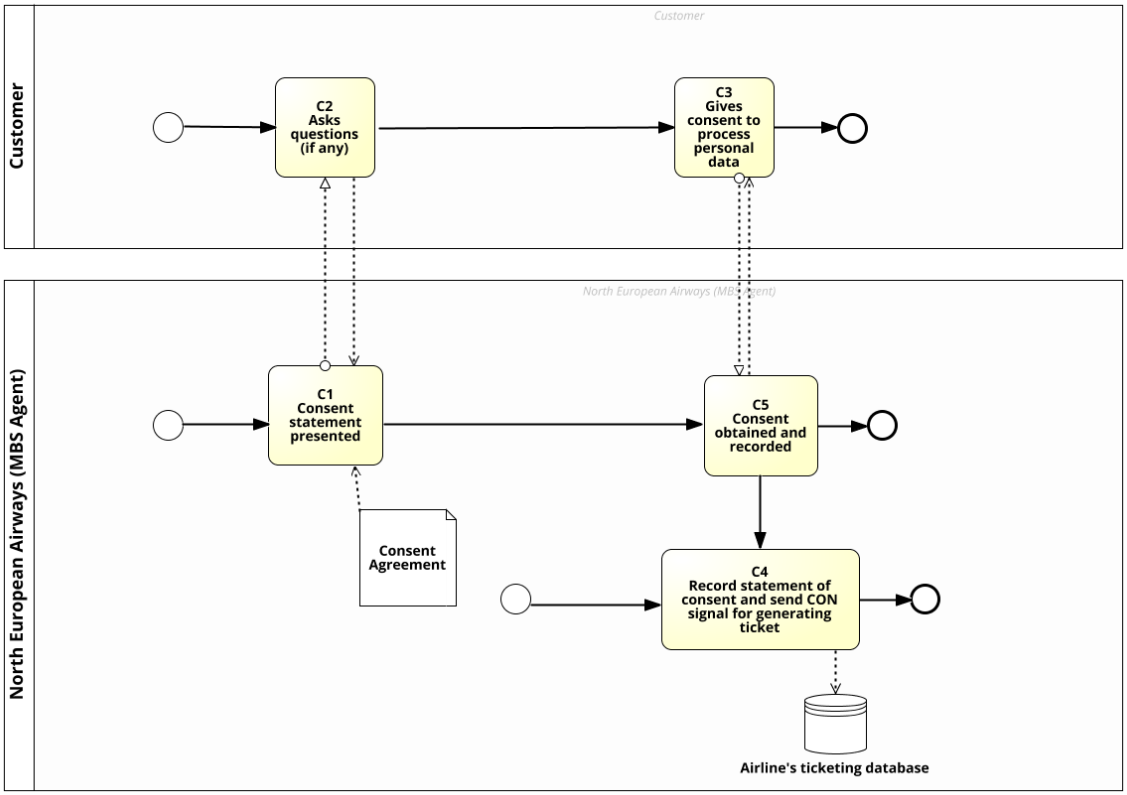


Figure 44. Consent Sub-Process Expert-Opinion Based Method

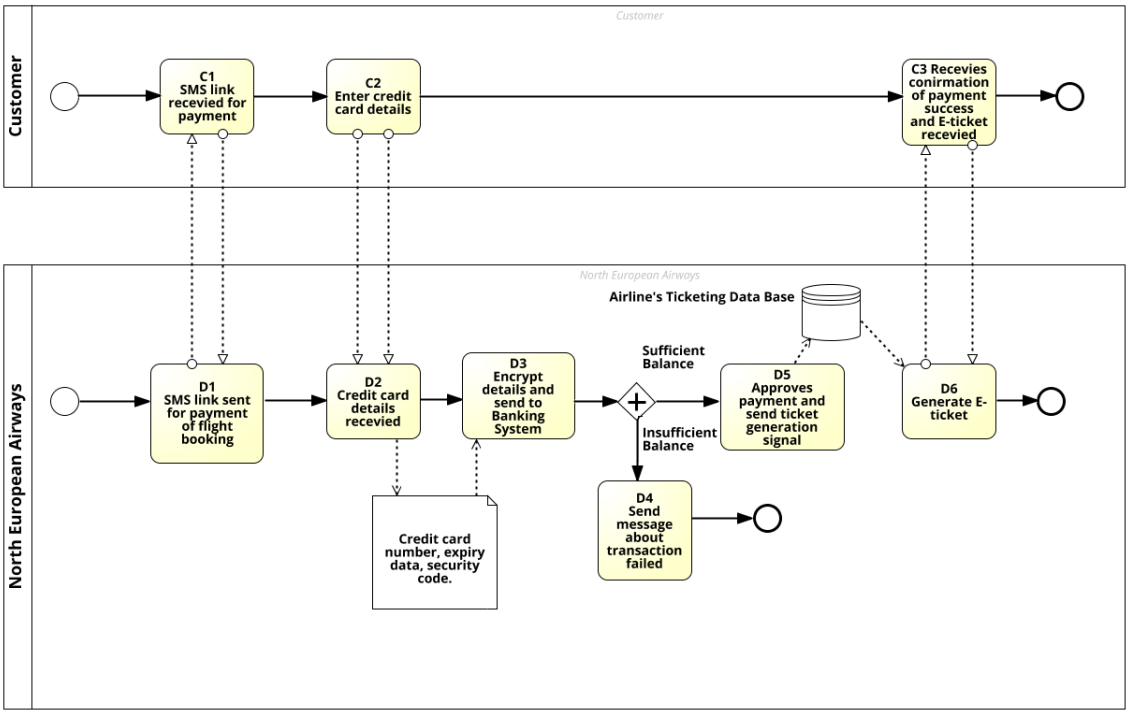


Figure 45. Expert-Opinion based Method to Collect Payment

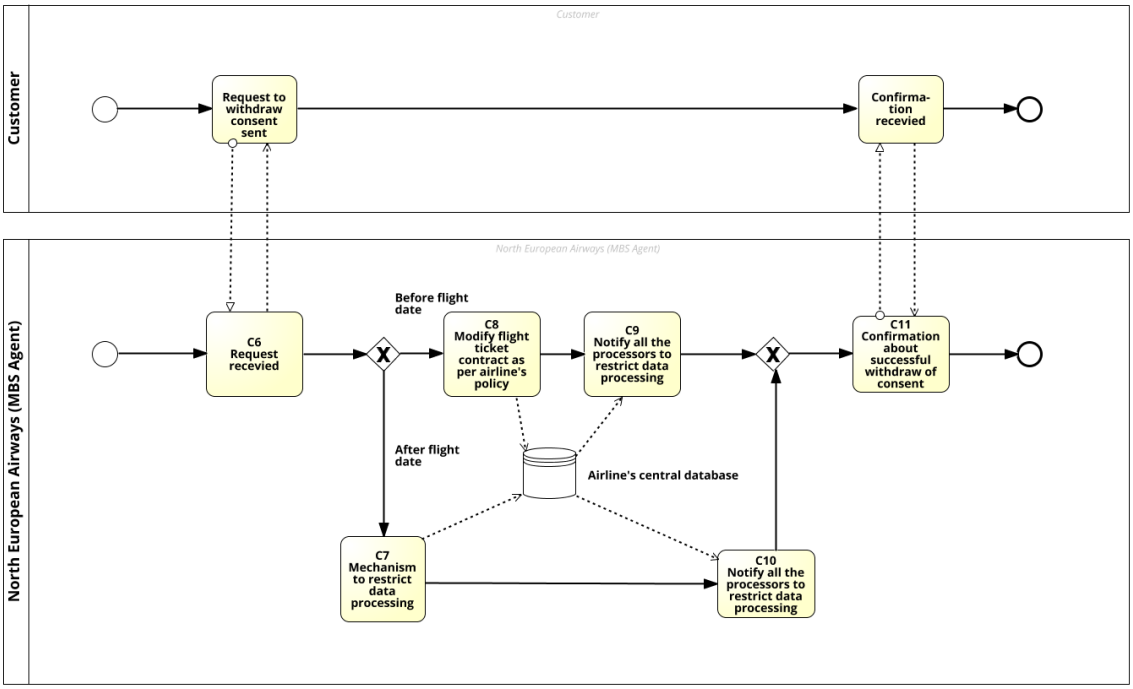


Figure 46. Consent Withdrawal Method Expert-Opinion Based Method

## VI. License

**Non-exclusive licence to reproduce thesis and make thesis public**

I, **Eduard Sing**,
(*author's name*)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis
**A Meta-Model Driven Method for Establishing Business Process Compliance to GDPR**,
*(title of thesis)*
supervised by Raimundas Matulevicius and Jake Tom,
*(supervisor's name)*

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **23.05.2018**