Automatically Proving Purpose Limitation in Software Architectures

Kai Bavendiek, Tobias Mueller, Florian Wittner, Thea Schwaneberg, Christian-Alexander Behrendt, Wolfgang Schulz, Hannes Federrath, Sibylle Schupp

Automatically Proving Purpose Limitation in Software Architectures

Kai Bavendiek[1], Tobias Mueller[2], Florian Wittner[3], Thea Schwaneberg[4], Christian-Alexander Behrendt[4], Wolfgang Schulz[3], Hannes Federrath[2], and Sibylle Schupp[1]

[1] Hamburg University of Technology (TUHH), Germany
[2] University of Hamburg (UHH), Germany
[3] Hans-Bredow-Institut for Media Research (HBI), Germany
[4] University Medical Center Hamburg-Eppendorf (UKE), Germany

Abstract. The principle of purpose limitation is one of the corner stones in the European General Data Protection Regulation. Automatically verifying whether a software architecture is capable of collecting, storing, or otherwise processing data without a predefined, precise, and valid purpose, and more importantly, whether the software architecture allows for re-purposing the data, greatly helps designers, makers, auditors, and customers of software. In our case study, we model the architecture of an existing medical register that follows a rigid Privacy by Design approach and assess its capability to process data only for the defined purposes. We demonstrate the process by verifying one instance that satisfies purpose limitation and two that are at least critical cases. We detect a violation scenario where data belonging to a purpose-specific consent are passed on for a different and maybe even incompatible purpose.

Keywords: medical register · GDPR · purpose limitation · compliance · data protection · privacy verification · software architectures

1 Introduction

Purpose limitation is a very relevant concept in the medical context where the adverse effects of misused data are arguably perceived more strongly and the requirement for privacy is not necessarily concerned with data ownership but much more with access and use of data [20]. The loss of confidentiality arises when the entity holding the patient's confidence conveys private information to another, unauthorised party. While it is hard to find documentations of medical studies that breached that confidence, "novel protocols for achieving confidentiality and security are urgently needed by the data mining community" [6]. Correspondingly, the GDPR recognises health data as particularly sensitive and in need of special protection by including it in Art. 9. On the other hand, medical registers are socially and politically accepted and must give access to researchers to serve their purpose.

The current practice of granting researchers access to medical data involves a complex and time-consuming process in which several boards and review committees are involved to make decisions. For example, the German Centre for Cardiovascular Research (DZHK) generously provides access to the data they have collected for several studies.[5] The process to get access to that data involves at least eight parties and has a lower bound of 10 weeks processing time, some of which is spent on determining whether privacy-related preconditions are met. A more automated process arguably increases the trust placed in the software system by developers, authorities, and users. Additionally, operators of a (medical) data collection platform will appreciate a proof of the system in which it is impossible to illegitimately collect or process data. Examples of the types of data collected by a medical platform to facilitate machine-assisted gathering of insights are age, sex, emergency department visits, office visits, comorbidity index, dyslipidemia, hypertension, cardiovascular disease, retinopathy, or end-stage renal disease [7]. Such data is typically collected for a broad and very general purpose in order to enable research to find associations of data that were not considered relevant in that context. The GDPR provides an exception[6] for collection of medical data without a specific purpose in such registers. However, participants of such a study might appreciate a proof of the limitation of the purposes of the data they provide.

CAPVerDE is a tool for designing architectures with privacy properties and generating above mentioned proofs. It uses formal methods to verify a given statement, such as "the register has access to personal data." We use CAPVerDE for modelling the architecture of GermanVasc, a real-world medical register for symptomatic peripheral arterial diseases [5]. GermanVasc follows a Privacy by Design approach. It encrypts the data as early as possible, implements access control through encryption [19], and limits the access to data to the minimum. We show how GermanVasc limits exposure of data based on the purpose researchers ask for obtaining access to the data. We demonstrate our privacy verification in the context of GermanVasc and show that it is practical.

More succinctly, the contributions of this papers are:

- a syntax and method for automatically verifying purpose limitation in the context of a software architecture,
- a software architecture for a real-world medical register which follows a Privacy by Design approach,
- and a case study in the context of a medical register demonstrating that automatically proving correct purpose limitation is feasible and practical.

This paper is structured as follows: In section 2 we give a brief legal background on the GDPR, before we present related work in section 3. Section 4 describes a medical register and section 5 a privacy verification logic. In section 6 we present the case study which we verify for purpose limitation violation in section 7. We discuss purpose limitation as well as our verification in section 8 and conclude the paper in section 9.

[5] Cf. `https://dzhk.de/en/research/clinical-research/use-and-access/`

[6] More on purpose limitation in the GDPR can be found in section 2.

2 Legal Prerequisites

The concept of the European General Data Protection Regulation (GDPR) demands every act of processing – be it collecting, storing, accessing, transmitting etc. – concerning personal data to be based on a legal ground in order to be lawful and legitimate. While Art. 6 (1) offers six different legal grounds, Art. 9 essentially limits it down to explicit consent given by the affected person (or "data subject") where especially sensitive data like genetic or biometric data or data concerning a person's health are concerned. Consent, like any other legal ground, then permits acts of processing for one or more specific purposes, usually set down in a privacy statement given to the data subject beforehand. The so-called principle of purpose limitation according to Art. 5 (1) (b) – "one of the stable bedrock principles in European data protection law" [11] – argues that any further acts of processing are generally only covered by this initial legal ground as far as they are necessary for the purpose(s) initially laid down. While this could potentially be bypassed by formulating a very broad initial purpose that could cover most future and unforeseen processing acts, the ideals of informed consent according to Art. 4 No. 11 and a specific and explicit purpose according to Art. 5 (1) (b) set strict boundaries.

Further processing for new and different purposes would consequently need a new legal basis, e.g. a new and adapted declaration of consent. However, exemptions from this strict limitation are possible. Art. 5 (1) (b), 6 (4) (a) state that an initial legal basis also covers acts of processing for those new purposes that are compatible with the initial purpose(s). Usually a complex matter of case-to-case consideration, the question of compatibleness in some cases, for example regarding acts of processing for scientific research and statistical purposes, is answered by the GDPR itself: according to Art. 5, they should "not be considered to be incompatible with the initial purposes." This means that, the implementation of appropriate technical and organisational measures pursuant to Art. 89 (1), e.g. pseudonymisation, provided, no new consent (or other legal ground) has to be sought in these cases [21].It is, however, important to note that this statutory privilege only covers those acts of processing for statistical purposes where the statistical results themselves are not personalised and are not used for future measures or decisions against specific persons, cf. recital 162.

3 Related Work

The need for preserving "privacy and security of human data" in the context of mining data from medical registers has already been acknowledged in 2002 [8]. It has also been subject of discussion in the context of genetic data which has the potential that future technologies can reveal much more about a patient than current technologies [16]. Consequently, storing medical data in a secure way has been subject of research [1,9,13]. While some of these works describe potential solutions, they did not actually implement any of them, let alone run an actual medical register with these techniques or cater for their particular needs with

giving access to yet unknown researchers. Neither do they discuss measures for limiting the purpose of acquired data or provide automatic proofs.

The importance of designing software in such a way that it respects Privacy by Design principles has been acknowledged several times [17,12,14]. In this work, we present a tool that can prove a system's capability to restrict the use of data for other purposes than they were collected for.

While to the best of our knowledge approaches of verifying purpose limitation in software architectures do not exist, previous work also used purpose hierarchies to formalize privacy. Fischer-Hübner and Ott already formalized purpose binding and necessity of data processing based on an access-control scheme [10]. Their approach considered different object classes, tasks, and purposes to determine the legality of data processing in the context of an enterprise. Karjoth, Schunter and Waidner used P3P[7]-like policies for an enterprise privacy management that is also based on access-control [15]. Their paper makes use of a purpose hierarchy to determine valid sub-purposes. Ashley et al. proposed the Enterprise Privacy Authorization Language (EPAL), which is a formal enterprise privacy policy language that also considers data types, hierarchies, and purposes. EPAL uses an XML-based syntax that aims at formalizing policies to achieve a machine-readable format. The above approaches all differ from the proposal of this paper because of their focus on enterprise-internal policies, while our goal is to describe and evaluate an "ecosystem" of potentially multiple actors (inter-enterprise) [2]. While the mentioned papers go in more detail about the different types of data processing, our description is more on a high-level without explicit details. This paper not only proposes a formal language but also presents a verification logic and a tool.

4 GermanVasc

A medical register following the Privacy by Design approach is GermanVasc [5]. It collects data about patients suffering from vascular diseases with the aim to assess the quality of treatments. To that end, the platform collects data about patients and their medical history. To increase quantity and diversity data it is designed to cater for over 50 study centres with up to 500 000 patients. This decentralised and distributed collection of data creates a challenge for the register [4].

In order to implement Privacy by Design the personal data of the patients are encrypted in a way that only allows the collecting entity, i.e. the hospital, to access the data. In particular, neither other hospitals nor the register itself have access to the key required for accessing the encrypted content. Similarly, medical data is stored encrypted with the encryption key stored on a separate medium, such that even a stolen database does not reveal any data, except for the metadata the database management system requires, e.g. row IDs. Taking the Privacy by Design approach further, researchers only get access to a subset

[7] P3P is an outdated website policy protocol with B2C focus.

$$
\begin{array}{lll}
A & ::= \{R\} & \\
R & ::= Has_i(\tilde{X}) & \mid Rec_{i,j}(\mathcal{P}, \{\tilde{X}\}) \\
 & \quad \mid Compute_i(\tilde{X} = T) & \mid Dep_i(\tilde{Y}, \{\tilde{X}^1, ..., \tilde{X}^n\})
\end{array}
$$

Table 1: Reduced syntax of architecture language

of the available medical data after a request has been granted by the register. In order for the register to make a decision about the requested access it can take the purpose into the account, which the researchers used when making the request. The register then needs to decide whether the researcher can get access to the data.

5 CAPVerDE

CAPVerDE is an acronym for Computer-Aided Privacy Verification and Design Engineering and refers to a project that includes a formal description and verification framework for privacy properties in software architectures as well as a tool that automatically performs this verification. For this paper we have enhanced the CAPVerDE formal verification language by adding syntax and semantics for defining purpose limitation. In the following sections we will focus on our additions rather than the basics, which are explained in [3], and hence give only a brief description of the necessary syntax and new semantics.

5.1 Syntax

The architecture language is the formal description of software systems' architectures with a focus on data flow and information flow between different components, e.g. representing actors of the system. An architecture consists of a set of relations that represent the data flow and information flow. Table 1 shows the relevant syntax of the architecture language. Again, we refer to [3] for more details.

The relation $Has_i(\tilde{X})$ models an entry point for the datum $\tilde{X}$ into the system, e.g. a measurement via a sensor of the component C_i. The new relation $Rec_{i,j}(\mathcal{P}, \{\tilde{X}\})$ represents the transmission of data with a purpose attached, also called a purpose-receive. Component C_i receives a set of variables $\{\tilde{X}\}$ from component C_j for the explicit purpose $\mathcal{P}$. $Compute_i(\tilde{X} = T)$ represents the ability of component C_i to compute a new variable $\tilde{X}$ from a term. Apart from the relations modelling the explicit flow, there are also relations that model the implicit data and information flow of a system. The dependence relation $Dep_i(\tilde{Y}, \{\tilde{X}^1, ..., \tilde{X}^n\})$ represents the ability of component C_i to obtain variable $\tilde{Y}$ when in possession of a set of variables $\{\tilde{X}^1, ..., \tilde{X}^n\}$.

The property logic, next, is the formal language to express properties that an architecture should satisfy. While the architecture language describes what is, the property logic describes what should be. Currently, it supports four different types of properties: the data a component can access, the knowledge it can gain, the data it shares with a third party, and the data it stores. In this paper we extend the properties by a fifth type that regards purpose violation. We only describe this new property. The property $notPurp_i$ represents the fact that component C_i does not comply with its purpose limitation. That is, it violates the property to only use purpose-restricted data for a compatible purpose and to only pass said data on to components with a compatible purpose. If an architecture satisfies this property, this means a purpose limitation violation.

5.2 Semantics

The semantics are based on states of components and events and traces. Additionally, a purpose role hierarchy and the mapping of a labelling function are necessary for our extension. The inverse labelling function $\mathcal{L}^{-1}$: $String \rightarrow Var^n,\ \mathcal{P} \mapsto \{\tilde{X}\}$ maps from purposes to sets of variables. The purpose hierarchy is derived from user input and explicitly states the partial order relations of all purposes of an architecture. To track the purposes attached to variables, we introduce the purpose state $State_P$. It assigns a purpose to each variable and is defined as follows: $State_P = (Var \rightarrow Purp)$, where $Purp$ is a purpose role that is a label for sets of variables. The initial state of the purpose state is $(X : \bot \,|\, \forall X \in Var)$. Each variable gets assigned $\bot$ that symbolises the bottom of the purpose role hierarchy with $\mathcal{L}^{-1}(\bot) = \emptyset$.

In this paper we only show our purpose-limitation-related changes to CAP-VerDE. When receiving variables with a purpose attached, this purpose role is stored in the purpose state of the receiving component. Purposes do not get lost when variables are passed on or altered. For example, if a component computes a new variable, the intersection of all variables' purposes is the purpose of the new variable. This is the conservative approach to prevent problems like aliasing. When a variable that has a purpose attached is received without a specific purpose, the original purpose is passed on to the receiving component. When a component deletes a variable from its local storage, the corresponding purpose is reset.

The semantics of the original properties including soundness and completeness proofs can be found in the mentioned general paper [3]. Here we present the semantics of the purpose limitation property $notPurp_i$:

$$A \in S(notPurp_i) \Leftrightarrow \exists \sigma \in \mathcal{S}(A),\, \exists \tilde{X} \in \sigma_i^v,\, \tilde{X} \notin \mathcal{L}^{-1}(\sigma_i^p(\tilde{X})) \vee (\sigma_i^o(\tilde{X}) = j \wedge \sigma_i^p(\tilde{X}) \not\sqsubseteq \sigma_j^p(\tilde{X}))$$

This expresses that an architecture A satisfies the property $notPurp_i$ iff a state σ of the architecture exists in which the component C_i has at least one

$$\mathbf{I}\wedge\frac{A \vdash \phi_1 \quad A \vdash \phi_2}{A \vdash \phi_1 \wedge \phi_2} \qquad \mathbf{I}\neg\frac{A \nvdash \phi}{A \vdash \neg\phi} \qquad \mathbf{P1}\frac{Rec_{i,j}(\mathcal{P}, E) \in A \quad \exists \tilde{X} \in E, \tilde{X} \notin \mathcal{L}^{-1}(\mathcal{P})}{A \vdash notPurp_i}$$

$$\mathbf{P2}\frac{Rec_{i,j}(\mathcal{P}, E) \in A \quad Rec_{k,i}(\mathcal{R}, F) \in A \quad \exists \tilde{X} \in E, \exists \tilde{Y} \in F, Dep_i(\tilde{Y}, Z), \tilde{X} \in Z \quad \mathcal{R} \not\sqsubseteq \mathcal{P}}{A \vdash notPurp_i}$$

Fig. 1: Verification rules regarding the purpose limitation property

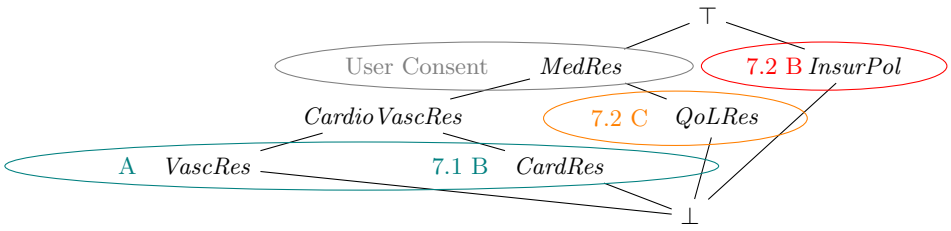

Fig. 2: Purpose hierarchy in a lattice structure

variable $\tilde{X}$ that either does not match its purpose or was received via an incompatible purpose. S is the semantics function and $\mathcal{S}$ denotes the set of all possible states of an architecture. σ_i^v, σ_i^p, and σ_i^o are the variable state, the purpose state and the origin state, respectively.

In the following we present verification rules that the CAPVerDE tool uses to verify properties in the context of an architecture. These rules are in the form of inference rules and a relevant excerpt is depicted in fig. 1. Rule $\mathbf{I}\wedge$ expresses that if component C_i receives variables with a purpose and not all the variables are covered by said purpose, the property $notPurp_i$ holds (illegitimate receiving). Rule **P2** states that if component C_i receives variables with a purpose $\mathcal{P}$ and then passes on part of these variables (or derivations) with a purpose $\mathcal{R}$ and $\mathcal{R}$ is not a sub-purpose of $\mathcal{P}$, then the property $notPurp_i$ holds (incompatible purpose). Rules $\mathbf{I}\wedge$ and $\mathbf{I}\neg$ regard the conjunction and negation of properties, respectively.

6 Case Study

In this section we introduce a case study which regards two cases: The positive case with valid use of the data and the negative case with illegitimate use of the

Purpose	MedRes	CardioVascRes	QoLRes	VascRes	CardRes	Profiling
Variables	$\{mD\}$	$\{emD\}$	$\{depression, salary, erectiledysfunction\}$	$\{bloodpressure, bloodcellcount\}$	$\{bloodpressure, cholestorellevel\}$	$\{pmD\}$

Table 2: Mapping of purposes and data types as variables

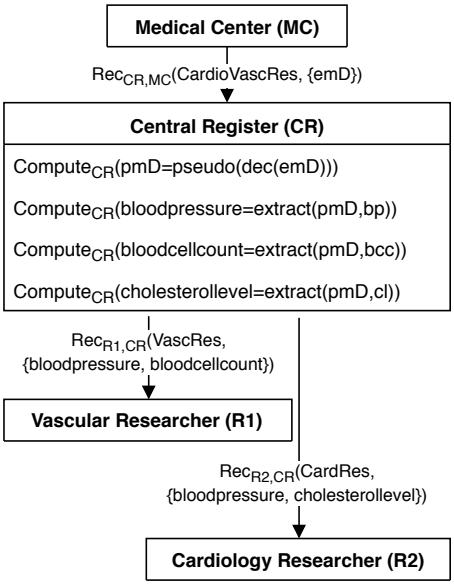

Fig. 3: Architecture for positive case

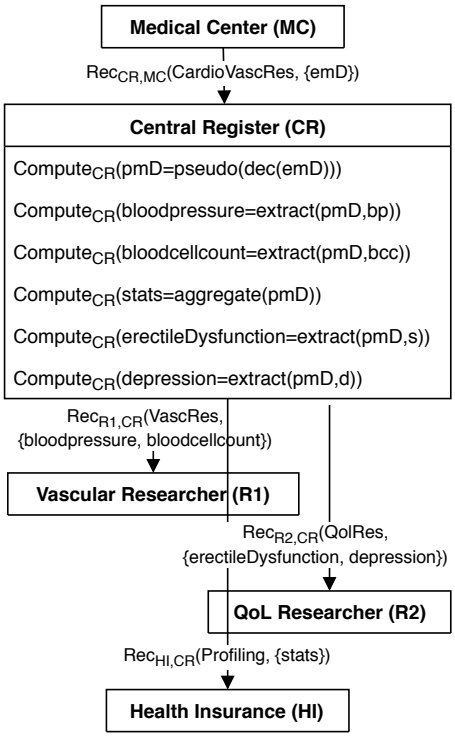

Fig. 4: Architecture for negative case

data. We will use CAPVerDe to check for purpose limitation violations in the next section.

Our medical data register case study is based on the real-world register GermanVasc as described in section 4. Our focus is on the purpose limitation aspect, therefore we omit a detailed description of the macro view and present only the relation between the register and the surrounding actors. Figure 3 shows a graphical representation of this view of the architecture following the syntax described in section 5. We have merged some components of the architecture which are not relevant for our purpose limitation case study such that it leaves only four to five actors.[8]

1) The positive case A medical center has the encrypted medical data it acquired from treating its patients. Because it wants to make this data available for further research, it acquires the patients' consent pursuant to Art. 6 (1) (a), 9 (2) (a) GDPR for transmitting the encrypted data to a central register for the purpose of cardiovascular research by researchers following this purpose. Architecturally phrased this means that the medical center *MC* sends the encrypted medical

[8] We make the full description of the architecture available under: `https://www.tuhh.de/sts/research/data-protection-machine-learning/capverde.html`

data *emD* with the attached research purpose *CardioVascRes* to the central register *CR*. The purpose hierarchy presented in fig. 2 demonstrates how sub-purposes may be included in a more general purpose role. The gray area marks the entry level of consent the user has given. Table 2 shows the relation between the purpose roles and the corresponding data types, i.e. the variables that can be received. The highlighted areas will be relevant in the following section. For this example the variable *emD* only contains the necessary information for the two purposes of vascular research and cardiology research. The central register can decrypt and pseudonymise the medical data to obtain the pseudonymised medical data *pmD*. From this the data for the vascular disease researcher $R1$ and the cardiological diseases researcher $R2$ can be extracted. The researcher with the purpose of vascular diseases *VascRes* receives the *bloodcellcount* (abbreviated as *BCC*) and the *bloodpressure* (abbreviated as *BP*), while the other researcher with the purpose of cardiology research *CardRes* receives information about the *bloodpressure* and the *cholesterollevel* (abbreviated as *CL*).

In addition to the explicit relations of an architecture, the software designer has to model the dependence relations. For this architecture we have $Dep_{CR}(pmD, \{emD\})$, $Dep_{CR}(BP, \{pmD\})$, $Dep_{CR}(BCC, \{pmD\})$, and $Dep_{CR}(CL, \{pmD\})$. Additionally, the architecture includes all dependence relations deduced from transitivity, i.e. $Dep_i(\tilde{Y}, \{\tilde{X}\}) \wedge Dep_i(\tilde{Z}, \{\tilde{Y}\}) \implies Dep_i(\tilde{Z}, \{\tilde{X}\})$, and also reflexivity, i.e. $Dep_i(\tilde{X}, \{\tilde{X}\})$, for all components C_i.

2) The negative case The slightly altered architecture depicted in fig. 4 shows a different scenario with two new actors. The second researcher is replaced by a quality of life (QoL) researcher and a health insurance actor comes into play. The central register now shares the statistical data *stats* with the insurance company *HI* attached with the purpose of policy profiling *Profiling*. The medical data of the research register is used for improving the quality of the insurance predictions in order to adapt the conditions for individual insurance members depending on how the predictions compare to their characteristics. The QoL researcher receives information about *depression* (abbreviated as *D*) and *erectiledysfunction* (abbreviated as *ED*) about the patients with the purpose of QoL research *QoLRes*. The purpose role hierarchy in fig. 2 stays unchanged and similarly table 2 is still valid for the second architecture. However, the variable *emD* now also contains information about depression and erectile dysfunction.

For the second architecture we have the following additional dependence relations: $Dep_{CR}(ED, \{pmD\})$, $Dep_{CR}(D, \{pmD\})$, and $Dep_{CR}(stats, \{pmD\})$. Again, we assume transitivity and reflexivity.

7 Verification

In the following subsection we demonstrate how to verify whether the described architecture satisfies the purpose limitation properties. The property described in section 5 checks two aspects: first, whether the purpose-receive relation itself was valid and second, whether the "purpose-chain" was valid. In the presented syntax

the property that expresses that an actor has violated its purpose limitation can be expressed as $notPurp_i$. The verification is done automatically by CAPVerDE and we demonstrate the steps of this verification process by tracing the algorithm in the following sub-sections.

Dual-Researcher Example In our first case of the architecture depicted in fig. 3 we want to verify whether any of the three actors, namely the central register and the two researchers, has violated the principle of purpose limitation by processing the medical data for a purpose that is neither covered by nor compatible with the purpose declared in the patients' original consent. Therefore, we want to verify the property: $\neg notPurp_{CR} \wedge \neg notPurp_{R1} \wedge \neg notPurp_{R2}$.

To do this CAPVerDE pattern-matches to apply the rules presented in fig. 1 and backtracks if necessary. The verifier tries to apply the axioms presented in fig. 1. As the property is a conjunction of properties, it applies the conjunction rule $\mathbf{I}\wedge$ twice. Therefore, the verifier checks all three sub-properties in turn and only if all three hold, the conjunction property holds. As all sub-properties are negations, it applies the negation rule $\mathbf{I}\neg$. Hence, the negated properties must not hold for the whole property to hold. Starting with the first (negated) sub-property $notPurp_{CR}$ CAPVerDE tries to apply the corresponding rules **P1** and **P2**. If either one is applicable, the property holds. By looking at rule **P1**, it can pattern-match the purpose-receive with the relation $Rec_{CR,MC}(CardioVascRes, \{emD\})$. Thus, the verifier needs to check whether emD is not part of the purpose variables of $CardioVascRes$. A look at table 2 shows that encrypted medical data are valid for this purpose. Hence, rule **P1** is not applicable. If we look at the second rule (**P2**), CAPVerDE can pattern-match the first purpose-receive, again, with the relation $Rec_{CR,MC}(CardioVascRes, \{emD\})$. There are two options for matching the second one, with either $Rec_{R1,CR}(VascRes, \{BP, BCC\})$ or $Rec_{R2,CR}(CardRes, \{BP, CL\})$. The verifier then considers the purpose-receive between the register and the vascular researcher first. In this case we have $E = \{emD\}$ and $F = \{BP, BCC\}$ as well as the purposes $\mathcal{P} = MedRes$ and $\mathcal{R} = VascRes$. CAPVerDE can deduce that $\tilde{X} = emD$. There are, in fact, the two transitive dependence relations $Dep_{CR}(BP, \{emD\})$ and $Dep_{CR}(BCC, \{emD\})$. Therefore, the verifier needs to check whether the partial order relation $VascRes \not\sqsubseteq CardioVascRes$ holds. A look at fig. 2 shows that the latter purpose is annotated as **A**. The relation does not hold because $CardioVascRes$ is an upper bound for $VascRes$. Thus, rule **P2** does not apply for the first researcher branch. Let us now take a look at the purpose-receive between the register and the cardiology researcher. $\mathcal{R}$ changes to $CardRes$ and F changes to $\{BP, CL\}$. The dependence relation $Dep_{CR}(CL, \{emD\})$ exists, so the verifier has to check the partial order constraint $CardRes \not\sqsubseteq CardioVascRes$. Figure 2 depicts $CardRes$ as case **7.1 B**. This relation does not hold neither, because $CardioVascRes$ is also an upper bound for $CardRes$. Therefore, rule **P2** is not applicable for the second researcher branch, neither. Hence, CAPVerDE can deduce that the sub-property $notPurp_{CR}$ does not hold and that thus its negation does.

The verification of the remaining sub-properties works in a similar fashion and is, therefore, omitted. The two properties $notPurp_{R1}$ and $notPurp_{R2}$ also do not hold. As, thus, all three (negated) sub-properties of the conjunction hold, the conjunction property also does. Therefore, CAPVerDE has successfully verified the full property $\neg notPurp_{CR} \wedge \neg notPurp_{R1} \wedge \neg notPurp_{R2}$ that states that none of the three actors violate their purpose limitation.

Health Insurance Example In our second case of the architecture, depicted in fig. 4, the verifier proceeds in a very similar way. We want to verify that the three actors, namely the central register, the researcher, and the insurance, do not violate their respective purpose limitation property. Hence, the property $\neg notPurp_{CR} \wedge \neg notPurp_{R1} \wedge \neg notPurp_{R2} \wedge \neg notPurp_{HI}$ must hold for this.

We omit the verification of conjunctions and negations and only demonstrate the verification of the four sub-properties: $notPurp_{CR}$, $notPurp_{R1}$, $notPurp_{R2}$, and $notPurp_{HI}$. As neither the branch of the Vascular Researcher has changed nor have the purpose role hierarchy and the connected data types, we carry over the result from the previous verification. Thus, we omit this part of the verification. We also focus on the purpose-chain checks and therefore only present the verification of $notPurp_{CR}$ as we have shown the approach in the previous example. CAPVerDE checks the property for the register: $notPurp_{CR}$. The verification of the first rule is the same as in the previous example. Thus, we omit it and focus on rule **P2**. For the pattern matching of the outgoing purpose-receives we only consider the two new branches. The health insurance branch gives us $\mathcal{P} = CardioVascRes$ and $\mathcal{R} = Profiling$ and $E = \{emD\}$ and $F = \{stats\}$. The dependence relation $Dep_{CR}(stats, \{emD\})$ shows that the verifier has to check the purpose hierarchy. *Profiling* is highlighted in fig. 2 in red and annotated as **7.2 B**. The partial order relation $Profiling \not\sqsubseteq CardioVascRes$ does hold because, as fig. 2 shows, the join of the two purposes is $\top$. Therefore, rule **P2** is applicable and the property $notPurp_{CR}$ holds in this architecture. This seems to be a case of purpose limitation violation because profiling is not covered by the purpose defined in the patients' consent. As this is already a violation, the verifier would stop here. Therefore, we omit the verification of the QoL branch. In our chosen purpose hierarchy CAPVerDE also detects a violation in this branch (cf. fig. 2 annotated as **7.2 C**).

Therefore, our tool detects this case as a violation of purpose limitation due to the way we modelled the purposes and their compatibility. While these acts of processing still serve the overall purpose of research for furthering the medical treatment of certain illnesses, the patients gave consent for the narrower purpose of cardiovascular research. Therefore, the principle of purpose limitation seems to be, again, not preserved. However, in both cases the principle of purpose limitation could still be preserved. As our model only checks if the new purpose is still covered by the initial purpose, it does not make a statement about the question of compatibility. Further processing for the purposes of quality-of-life research and profiling could, at least potentially, be covered by the assumed-compatibility clause in Art. 5 (1) (b). Although our approach does not assume

such compatibility, one could argue that the "parent" of the two purposes is medical research and therefore the purpose *QoLRes* is not strictly incompatible with the initial purpose of the patient.

8 Discussion

Our contribution is a method and an extension to a tool for effectively applying a rigorous Privacy by Design approach manifested in the limitation of purpose of the collected data. As CAPVerDE is open-source software and the approach very generic, it is possible to apply the same method to other contexts. One limitation of the presented approach to automatically prove purpose limitation is the need for a correct model. If the model does not accurately reflect the data flows within the software, the verification cannot provide reliable information. Similarly, while the verifier can automatically answer questions, it is essential that the correct questions be asked. We have suggested a set of classifications and formulae for the specific domain of a medical register. This set is, obviously, not directly usable for other contexts but we envision that it can serve as a template for other domains.

Also we have to discuss the obvious trade-off between innovation and purpose limitation. This discussion has been had many times regarding the conflict between purpose limitation and big data analytics in general. While the GDPR does offer some relief in this regard with the compatibility clauses in Art. 5 (1) (b) it remains to be seen if this is enough to allow sufficient innovation. One reason to doubt that stems from the fact that due to several flexibility clauses like Art. 89 (2) regarding processing for (medical) research and statistical purposes, the details are up to the member states and their respective national laws. This fragmentation makes innovation difficult and contradicts the purpose of unification otherwise pursued by the GDPR [18].

From the technical aspect, our presented approach considers an inflexible purpose role hierarchy that prevents, for instance, new associations in the medical field that would attract further research. If we wanted to allow a new association, the purpose roles have to be re-evaluated and updated frequently. Here, the question of compatibility would need to be checked on two levels: firstly, regarding the pre-formulated assumptions of compatibility in Art. 5 (1) (b). And secondly, where none of the explicit purposes described there apply, a free-hand check taking into account and balancing the legitimate interests of both controller and data subject(s) as stated by Art. 6 (4). While this would arguably be an ambitious goal for an approach like ours, all parties could benefit from it.

The legal analyses performed in this paper are not thorough and we make no claims about the fitness of our method in court. We refer to other papers for further discussion [4,5]. While we believe that designing and analysing software systems with our method provides value to the stakeholders, we do not and cannot make any further statements.

9 Conclusion and Future Work

This paper presents a method for automatically verifying purpose limitation in the context of software architectures to comply with the principle of Privacy by Design. The paper also describes a real-world example of a privacy-aware medical register. We use this example as a case study to demonstrate our proposed method. We model the architecture of said register and verify formal purpose limitation properties in two example instances. The paper presents a brief legal discourse on purpose limitation in the context of GDPR and its value in practice. We have shown how Privacy by Design can be implemented with regards to purpose limitation and hope to inspire the modelling of future applications.

Art. 5 of the GDPR names seven "principles relating to processing of personal data." Art. 25 (1), laying down the ideal of privacy by design within the GDPR, accordingly obliges controllers to "implement appropriate technical and organizational measures" to further these principles. The extent of this obligation is subject to, inter alia, the "state of the art." This reflects the GDPR's general ideal of openness to development; an ideal that leaves controllers quite a bit of leeway regarding the "how" of the fulfilment of their obligations and provides them with the possibility of developing solutions and tools that prevail themselves as industry-wide standards. Until enough of these standards have been established, though, the obligation's vagueness and openness lacks the clarification and guidance needed for most controllers. We hope that our tool can contribute to the concretisation of this notion. In this paper we have focussed on purpose limitation. The different principles are closely related. Data minimisation and storage limitation are two principles that can, to a certain extent, already be expressed with the proposed tool. The accountability mentioned in Art. 5 (2) is a new aspect worth looking into. A tool-assisted approach to demonstrate compliance is an interesting field for future work. Similarly, an automated process for the "data protection impact assessment" described in Art. 35 is a promising direction for further research. Finally, it is worthwhile exploring to what extent the management of purposes can be automated.

Acknowledgement The work is part of the Information Governance Technologies project which is funded by the Behörde für Wissenschaft, Forschung und Gleichstellung.

The IDOMENEO study is funded by the German Joint Federal Committee (Gemeinsamer Bundesausschuss, G-BA) (01VSF16008) and by the German Stifterverband as well as by the CORONA foundation (S199/10061/2015).

References

1. Akinyele, J.A., Lehmann, C.U., Green, M.D., Pagano, M.W., Peterson, Z.N.J., Rubin, A.D.: Self-Protecting Electronic Medical Records Using Attribute-Based Encryption. Tech. Rep. 565 (Nov 2010)
2. Ashley, P., Hada, S., Karjoth, G., Powers, C., Schunter, M.: Enterprise Privacy Authorization Language (EPAL). IBM Research **30** (2003)

3. Bavendiek, K., Adams, R., Schupp, S.: Privacy-Preserving Architectures with Probabilistic Guaranties. In: Proceedings of the 16th International Conference on Privacy, Security and Trust. pp. 1–10. IEEE (Aug 2018)
4. Behrendt, C.A., Ir, A.J., Debus, E.S., Kolh, P.: The Challenge of Data Privacy Compliant Registry Based Research. European Journal of Vascular and Endovascular Surgery **55**(5), 601–602 (May 2018)
5. Behrendt, C.A., Pridöhl, H., Schaar, K., Federrath, H., Debus, E.S.: Klinische Register im 21. Jahrhundert. Der Chirurg **88**(11), 944–949 (Nov 2017)
6. Berman, J.J.: Confidentiality issues for medical data miners. Artificial Intelligence in Medicine **26**(1), 25–36 (Sep 2002)
7. Breault, J.L., Goodall, C.R., Fos, P.J.: Data mining a diabetic data warehouse. Artificial Intelligence in Medicine **26**(1), 37–54 (Sep 2002)
8. Cios, K.J., William Moore, G.: Uniqueness of medical data mining. Artificial Intelligence in Medicine **26**(1), 1–24 (Sep 2002)
9. Drosatos, G., Efraimidis, P.S., Williams, G., Kaldoudi, E.: Towards Privacy by Design in Personal e-Health Systems. In: Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies. vol. 5, pp. 472–477. Rome (Feb 2016)
10. Fischer-Hübner, S., Ott, A.: From a Formal Privacy Model to Its Implementation. In: Proc 21st Nat Information Systems Sec Conf (1998)
11. Forgó, N., Hänold, S., Schütze, B.: The Principle of Purpose Limitation and Big Data. In: Corrales, M., Fenwick, M., Forgó, N. (eds.) New Technology, Big Data and the Law, pp. 17–42. Perspectives in Law, Business and Innovation, Springer Singapore, Singapore (2017)
12. Graf, C., Wolkerstorfer, P., Geven, A., Tscheligi, M.: A Pattern Collection for Privacy Enhancing Technology (Jan 2010)
13. Haas, S., Wohlgemuth, S., Echizen, I., Sonehara, N., Müller, G.: Aspects of privacy for electronic health records. International Journal of Medical Informatics **80**(2), e26–e31 (Feb 2011)
14. Hafiz, M.: A pattern language for developing privacy enhancing technologies. Software: Practice and Experience **43**(7), 769–787 (Jul 2013)
15. Karjoth, G., Schunter, M., Waidner, M.: Platform for Enterprise Privacy Practices: Privacy-Enabled Management of Customer Data. In: Int Workshop on Privacy Enhancing Technologies. pp. 69–84. Springer (2002)
16. Kaye, J., Boddington, P., de Vries, J., Hawkins, N., Melham, K.: Ethical implications of the use of whole genome methods in medical research. European Journal of Human Genetics **18**(4), 398–403 (Nov 2009)
17. Kung, A.: PEARs: Privacy Enhancing ARchitectures. In: Preneel, B., Ikonomou, D. (eds.) Privacy Technologies and Policy. pp. 18–29. Lecture Notes in Computer Science, Springer International Publishing (2014)
18. Mayer-Schönberger, V., Padova, Y.: Regime Change? Enabling Big Data through Europe's New Data Protection Regulation. The Columbia Sciene & Technology Law Review **17**(315), 21 (May 2016)
19. Pilyankevich, E., Korchagin, I., Mnatsakanov, A.: Hermes. A framework for cryptographically assured access control and data security. Tech. Rep. 200 (Feb 2018)
20. Safran, C., Bloomrosen, M., Hammond, W.E., Labkoff, S., Markel-Fox, S., Tang, P.C., Detmer, D.E.: Toward a National Framework for the Secondary Use of Health Data: An American Medical Informatics Association White Paper. Journal of the American Medical Informatics Association **14**(1), 1–9 (Jan 2007)
21. Schulz: DS-GVO Art. 6 Rechtmäßigkeit der Verarbeitung. Gola p. 210 (2018)