Management of Customized Privacy Policies

Jens Leicht[a] and Maritta Heisel[b]
Paluno - The Ruhr Institute for Software Technology, University of Duisburg-Essen, Germany
{*jens.leicht, maritta.heisel*}*@uni-due.de*

Keywords: Privacy Policy Management, Consent Management, Privacy Policy Customization, Privacy Policy Storage, Data Value Chain, Data Protection Legislation, Data Accountability.

Abstract: While privacy policies are well established to express data processing practices, customizable privacy policies are a researched but not established practice to empower data subjects. One of the hurdles, hindering the acceptance of customizable policies, is the management of large amounts of privacy policies, when each data subject has their own policy. We propose a Privacy Policy Management (PPM) system, which handles customized policies and distributes them to all data processors. In addition, our PPM keeps track of where and why data are being transferred. This information can be provided to the data subjects, so that they can see that the data controller complies to the policy agreed upon. The log of data transfers can also be used by data protection authorities, to check the GDPR-compliance of the data controller or for investigations in case of a data breach. We discuss the architecture of our PPM, how it operates, and integrate it into the Privacy Policy Compliance Guidance framework.

1 INTRODUCTION

Privacy policies are an important tool for service providers to be able to comply with data protection legislation, like the General Data Protection Regulation (GDPR) of the European Union (European Parliament and Council of the European Union, 2016). These policies inform the data subjects about the data processing practices of the data controller and are used to collect data subjects' consent. While there has been development towards customizable privacy policies with, e.g., the Platform for Privacy Preferences (P3P) or the PriPoCoG-framework, it remains open how data controllers shall handle the large amount of different customized privacy policies (Cranor et al., 2006; Leicht et al., 2022). From this gap, we derived the following research questions:

RQ1. How can data controllers and data processors be supported, when using customizable privacy policies?

RQ2. How can data flows be made more transparent towards the data subjects?

To address the research questions, we propose a Privacy Policy Management (PPM) system, that supports data controllers and processors in handling customized privacy policies. We consider not only the data controllers' storage of customized privacy policies, but also the different stages of the distribution of the policies. The privacy policies are transferred along the data value chain, so that all data processors know the agreed upon privacy policy and can adhere to customized policies. Our PPM manages policy customization by the data subject, as well as policy updates by the data controller itself. We integrate our PPM into the PriPoCoG-framework. Future work is needed to implement and evaluate the proposed PPM.

In addition to the main purpose of policy management, our approach also increases the transparency towards the data subjects. This is achieved by providing data transfer logs, which clearly state where data is transferred to. Providing these logs to data subjects, in addition to the privacy policy, may increase their trust into the data controller. Furthermore, these logs can be used by data protection authorities to check the GDPR-compliance of the data controller, as well as in investigations in case of a data breach. The logs provide accountability of which data processor received data from which data subject.

Before we explain our privacy policy management system in detail, we provide a concise background in Section 2, introducing all necessary concepts. Next, we place the management system inside the PriPoCoG-framework in Section 3. In Section

[a] https://orcid.org/0009-0003-5612-5590
[b] https://orcid.org/0000-0002-3275-2819

Leicht, J. and Heisel, M.
Management of Customized Privacy Policies.
DOI: 10.5220/0013365300003899

In *Proceedings of the 11th International Conference on Information Systems Security and Privacy (ICISSP 2025) - Volume 2*, pages 385-396
ISBN: 978-989-758-735-1; ISSN: 2184-4356

4 we present our management concept, followed by related work in Section 5. Finally, we close with a conclusion and future work in Section 6.

2 BACKGROUND

In this section, we provide the necessary background information for a better understanding of our privacy policy management system.

2.1 General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) is a regulation of the European Union (EU) (European Parliament and Council of the European Union, 2016). It was enacted to protect the privacy of European citizens from unwanted data processing. The GDPR not only applies to businesses inside the EU, but also to anybody outside the EU who processes personal data of EU citizens. Hence, it is important for businesses, called *data controllers* in the GDPR, to be able to prove their GDPR-compliance by keeping track of consent provided by the end-users (*data subjects*).

Data minimization is one of the concepts of the GDPR. It aims to protect privacy by only allowing the minimum data collection and processing necessary to provide a service. The customizable privacy policies of the PriPoCoG-framework (Leicht et al., 2022) enable data controllers and data subjects to practice data minimization, while allowing them to include additional data processing if explicitly enabled by the data subjects (cf. Section 3).

Informed consent is one of the legal bases for data collection and processing, according to the GDPR. Data controllers need to collect this informed consent and need to be able to prove it towards data protection authorities. Our PPM helps data controllers to manage customized privacy policies and informed consent in a unified system.

2.2 Privacy Policy Life Cycle

The privacy policy life cycle presented in Figure 1 depicts the different phases of a privacy policy. The life cycle starts with the creation of the privacy policy. The second step is the visualization of the policy towards the data subjects, followed by the collection and management of data subjects' consent. When customizability is provided, we need to be able to visualize and customize the policy again, even after the initial consent has been collected. Hence, the arrow from consent management to policy visualization. Once the consent has been collected, the policy needs to be available along the data value chain, so that anybody handling the data can adhere to the privacy policy. The data value chain describes where data are transferred to and who may process it. In terms of the GDPR, it describes all *data processors* that handle the data on behalf of the data controller. The final step that the privacy policy has to manage is data retention, making sure that the data is not kept longer than is stated in the privacy policy. We highlight the phases *Consent Management* and *Data Value Chain* in bold, as these are the phases where our PPM comes into action.

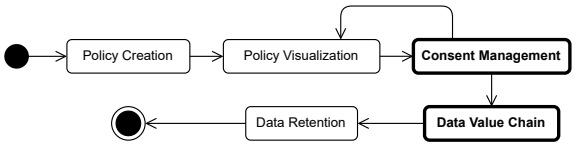

Figure 1: Privacy Policy Life Cycle, with steps relevant to our PPM highlighted in bold.

3 FRAMEWORK

The Privacy Policy Compliance Guidance (PriPoCoG) framework was developed to support the GDPR-compliance of data controllers by augmenting the privacy policy definition process with automated compliance checks (Leicht et al., 2022). Furthermore, the framework empowers data subjects in their privacy choices by making privacy policies customizable, similar to the options provided by cookie banners. Not only does the framework check compliance of the privacy policies created within it, but also supports policy authors in reusing information from the software development process for the definition of detailed and transparent privacy policies (Leicht et al., 2023).

Our Privacy Policy Management (PPM) embeds into the PriPoCoG-framework, as depicted in Figure 2. The PPM is highlighted in orange, and it connects with each of the other four components of the framework. Additionally, data subjects, as well as data protection authorities, can interact with the PPM. The other parties (policy authors and data processors) interact, via existing components of the framework, with the PPM.

In PriPoCoG, privacy policies are created using the *Privacy Policy Editor* (Leicht and Heisel, 2024). Using the *DFD-Editor*, the policy authors can import information regarding data flows from data-flow diagrams (DFDs) (Leicht et al., 2023). The policy,

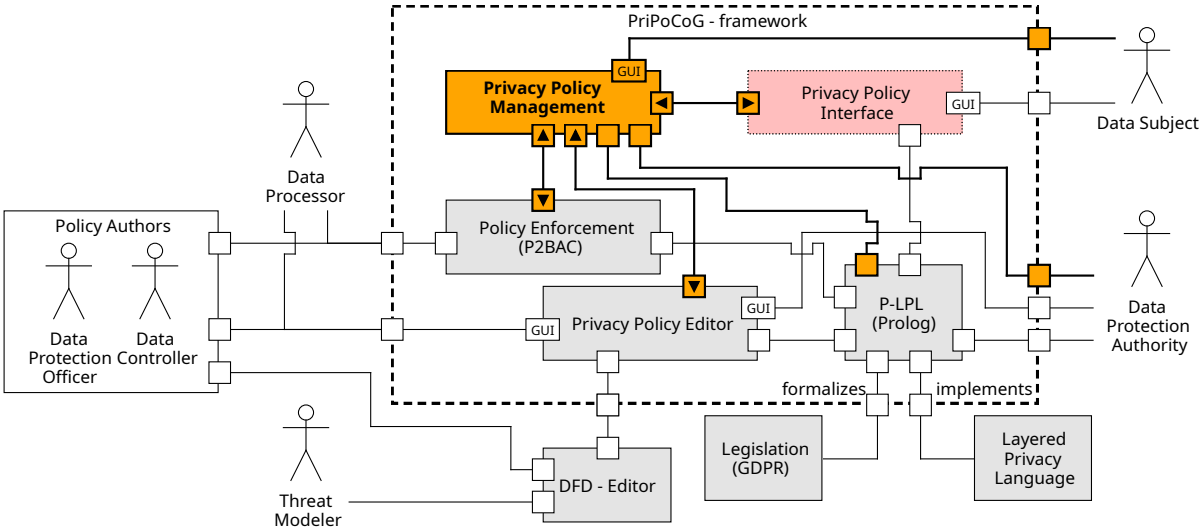

Figure 2: Overview of the PriPoCoG-framework with the PPM and its connections highlighted in **orange**; unpublished and open research highlighted in **pink**; based on (Leicht and Heisel, 2024, Figure 2).

when defined by the data controller, is stored in the PPM as a basis for the customized policies of the data subjects. Compliance checks are performed by P-LPL (Leicht et al., 2022), the Prolog - Layered Privacy Language, which is a formalized and extended version of the Layered Privacy Language by Gerl (Gerl, 2020) and the GDPR (European Parliament and Council of the European Union, 2016).

Since there is no publication regarding the *Privacy Policy Interface*, we assume that it collects the customized policies and consent from the data subjects. We connect the PPM to the interface to be able to process the customized policies. The PPM initially provides the original policy of the data controller to the data subject via the privacy policy interface. After the data subject customized the policy and provided at least partial consent, the customized policy is saved in the PPM. This actively customized policy is a proof of consent that can be provided to data protection authorities. The interface between the *Data Subject* and the PPM allows data subjects to retrieve information about which data processor received their data. This interface can for example be embedded in the privacy settings on the account settings page of the service. By providing this information to the data subject, the overall transparency is improved. The PPM can use the *P-LPL* backend to check privacy policies for GDPR-compliance and check the compatibility of customized policies with the original policy of the data controller. The customized privacy policies are enforced at the data controller and data processor side using Privacy Policy Based Access Control (*P2BAC*) (Leicht and Heisel, 2023). Data are collected from the data subject via the PPM, ensuring that policy enforcement is always used before accessing data. *Data Protection Authorities* can use the PPM to retrieve logs of data transfers, which can be used to further check the GDPR-compliance of the data controller. Additionally, these logs may be helpful in the event of a data breach, providing accountability of who received data from which data subject.

4 CONCEPT

In this section, we first describe the running example, which we use to explain our Privacy Policy Management (PPM). Next, we present the PPM features, followed by a more detailed look at the architecture and interaction between the different stakeholders and components involved in the PPM.

4.1 Running Example

Figure 3 shows an abstract data-flow diagram of our running example, an online shop. We do not detail which data are transferred or processed for the different sub-services.

The diagram is split into three trust boundaries: The *Online Shop* is the data controller; hence we highlight the trust boundary in red. The second trust boundary (to the left) represents a data processor, the *Shipping* company, which delivers the online shopping orders. Finally, the third trust boundary (to the right) represents a *Marketing* company, which processes the interests of the customer (data subject), to provide personalized advertisements. We use the two processes *Shopping* and *Advertising* to represent dif-

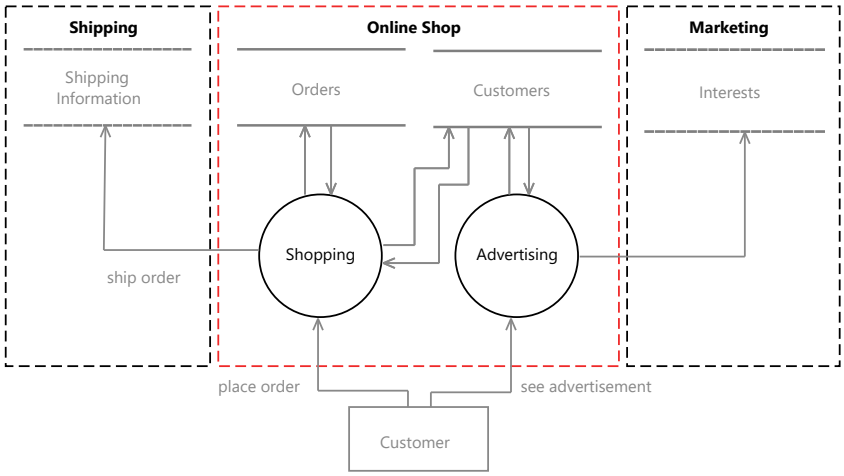

Figure 3: Abstract data-flow diagram of our running example "Online Shop".

ferent parts of the service provided by the online shop.

The customer can place orders in the shopping process, which requires customer data and produces order information, which are managed in the corresponding data storage (*Customers* & *Orders*). When the order is shipped, information like the delivery address is shared with the external data processor *Shipping*, who stores *Shipping Information*.

The second service provided by our online shop is personalized *Advertising*, which selects interest-based advertisements (ads) to be shown to the customer. The controller uses an external *Marketing* company for ad selection, which processes the interests of the customer.

Customers can prohibit the processing of their interests by customizing the privacy policy of the online shop. When they dissent the processing of their interests, no information will be forwarded to the marketing company. Instead, they will be prompted with randomly selected advertisements.

4.2 Features

Our proposed Privacy Policy Management (PPM) provides the following features:

- **F1** - Manage customized privacy policies.
- **F2** - Distribute customized privacy policies to data processors.
- **F3** - Manage policy updates by data controllers.
- **F4** - Manage policy customization by data subjects.
- **F5** - Collect data subjects' data according to the customized privacy policy.
- **F6** - Keep track of data processors.

The features F1 and F2 build the basis for the application of customized privacy policies in complex systems. F3 improves the process of updating privacy policies. Currently, a new legal document has to be defined and changes and corrections are bundled, before a new version of a policy is published. With our PPM, small corrections and changes to the policy can easily be deployed. Reacting to dynamic policy customization is an important feature (F4), especially (partial) withdrawal of consent. We propose that our PPM not only manages the policies, but also collects data from the data subjects (F5). This better protects data subjects from data misuse, as the data cannot be processed by bypassing the policy enforcement system. Finally, feature F6 provides accountability regarding data processors. The information tracked can increase transparency towards data subjects, if made accessible. Furthermore, data protection authorities can use this information for GDPR-compliance checks as well as data breach investigations.

4.3 Architecture

In this section we first discuss the privacy policy management architecture in general, before applying it to our running example.

4.3.1 General Explanation

Figure 4 shows the architecture of our Privacy Policy Management (PPM), explaining how it integrates with already existing systems of data controllers and data processors. The bold rectangles outline the boundaries of the PPM on the data controller's (left) and data processors' (right) side. The PPM consists of three components: the *Privacy Manager*, *Policy Storage*, and *Logging*. Both parties run their own instance of the PPM with similar functionalities. This ensures

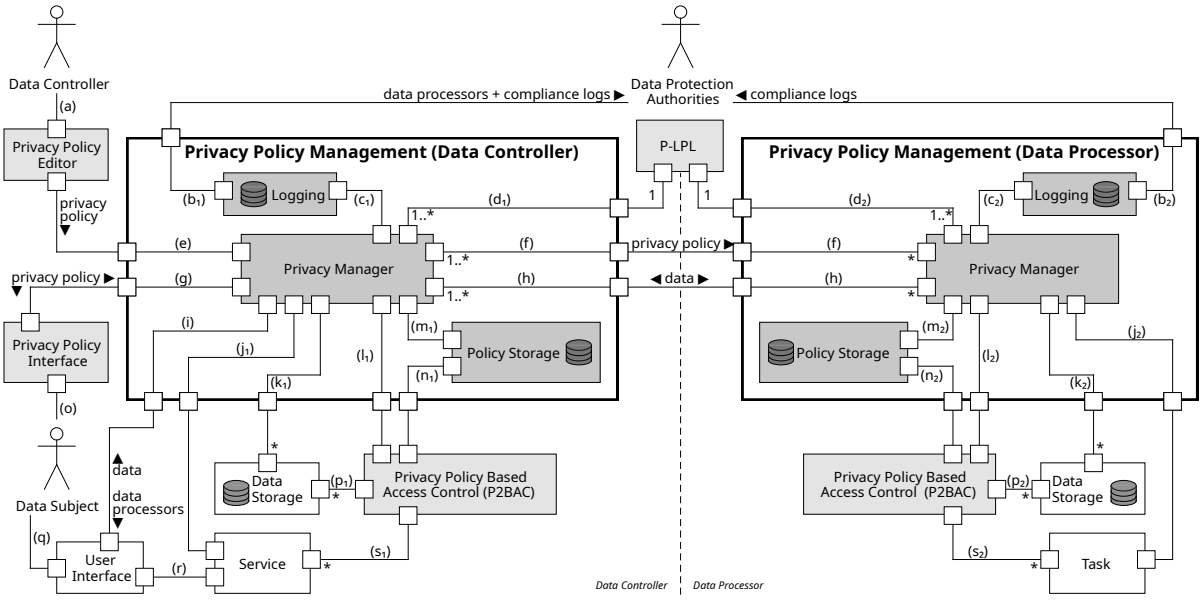

Figure 4: Architectural overview of the privacy policy management.

that all parties are able to manage customized policies locally. It also reduces the overhead and decision time of access requests, which would cause network traffic with a centralized PPM. Confidentiality would also be at risk, when using a more centralized PPM architecture. Only the data controller side directly interacts with the *Data Subject* (DS). Existing systems of the data controllers and data processors are visualized as white, and other PriPoCoG components as light gray rectangles. In the following, we reference interfaces from the architecture using their corresponding letters, e.g., *(a)*.

The *Data Controller* (DC) uses the *Privacy Policy Editor (a)*, which provides the defined privacy policy to the *Privacy Manager* (PM) via interface *(e)*. The DS customizes the privacy policy via the *Privacy Policy Interface (o)*, which forwards the resulting policy to the PM *(g)*. Data subject's data are collected via the *User Interface (q)* and forwarded to the PM via interface *(i)*. This separation of interfaces is necessary, as the privacy policy interface is not able to collect data from the DS, instead data are collected in various ways through the user interface of the service. The PM stores privacy policies and data in their corresponding databases (m_x) and (k_x). The DS interacts with the PM, retrieving a list of data processors, via the user interface of the service *(q)*, which is connected to the service via interface *(r)*.

The PM is responsible for transferring privacy policies and data to the data processors via *(f)* and *(h)*; not only initially, but also once a privacy policy is updated. This provides reliable intervenability, where all parties involved in processing data are informed about any withdrawn consent. The PM informs the Privacy Policy Based Access Control (P2BAC) about changes in the customized policy, via interface (l_x), which in turn can interrupt current data processing if consent has been withdrawn. We explain how policy updates are performed, in more detail in Section 4.4.

When the service requires some personal data, it requests access from the P2BAC system (s_1). P2BAC retrieves the corresponding policy from policy storage (n_1) and decides whether access will be granted. If the decision is positive, P2BAC grants the service access to the data (p_1) and (s_1). When the service derives new data concerning the DS, it provides this data to the PM via interface (j_1). The PM stores this new data according to the privacy policy in the corresponding data storage (k_1).

Similar to the service (DC), the task (Data Processor, DP) interacts with P2BAC and the PM (s_2), (n_2), (p_2), and (j_2). Interface (j_2) is also used to trigger tasks, when data processing is requested via the PM (cf. Section 4.4.2).

The PM keeps track of the data transfers it performs. It logs all data processors that received data from each DS and provides this information to the *Logging* component (c_1). On the DP side, the PM keeps track of the data controllers that provided data to the data processor (c_2). Logs are stored in databases for efficient access and exportability. The information from the logs is used, in case of a policy change by the DS, to inform all data processors about the updated policy. Additionally, this logging improves transparency towards the DS by allowing them to request a list of data processors *(i)* and *(q)*. The DS

can clearly see where their data are transferred and can compare the data transfers to the privacy policy they consented to. The logs can also be accessed by *Data Protection Authorities (b_x)* when examining the GDPR-compliance of a data controller or in case of a data breach. Keeping track of data processors also provides accountability regarding who is processing a data subject's data.

The PM can use *P-LPL*, via interface *(d_x)* to check privacy policies for GDPR-compliance and check the compatibility of customized policies with the original base policy of the data controller.

The interfaces *(f)* and *(h)* between data controller and data processor are many-to-many connections. A controller can have none, one, or many data processors, and a processor works for at least one controller. The policies only flow in one direction *(f)* from the controller to the processor. Since the processor may produce derived data, when performing its task, the data connection is bidirectional *(h)*.

4.3.2 Example

Applying this architecture to our example, we get the following instantiation. The *Online Shop* constitutes the data controller, and the *Shipping* company and the *Marketing* company are data processors. Each company has their own instance of the privacy policy management as well as a local instance of the Privacy Policy Based Access Control (P2BAC).

The service on the controller side is divided into the two sub-services: *Shopping* and *Advertising*. Each of these sub-services interact with P2BAC and the Privacy Manager (PM) to get access to data or store newly derived data. If the data subject (DS), for example, places an order, the service provides order details to the PM of our online shop, to be stored in the data storage.

If the DS consents to data processing by the shipping company, the PM will inform the shipping company for which purpose the DS allows the processing of their data. When an order is placed and ready to ship, the PM will provide the necessary and consented delivery details to the shipping company. The P2BAC of the shipping company ensures that data are only used for the consented purpose of delivering the order. Further processing by the shipping company, for example for marketing purposes, will be prohibited by P2BAC.

If the DS does not consent to personalized advertisements, the PM will not forward their privacy policy to the marketing company. It will also not forward any information regarding the DS to the marketing company. When the DS decides to enable personalized advertisements later on, the new privacy policy is provided to the marketing company by the PM of the online shop. Afterward, information about the interests can be forwarded and collected by the marketing company.

4.4 Modus Operandi

In the following we explain the modus operandi of the Privacy Policy Management in the two phases *Consent Management* and *Data Value Chain*. To be readable, the diagrams in this section abstract away some detailed operations.

4.4.1 Consent Management

Consent Collection. The interaction visualized in Figure 5 describes the initial collection of consent. The Data Subject (DS) uses the Privacy Policy Interface, which requests the privacy policy, defined by the policy authors, from the Privacy Manager (PM) on the Data Controller (DC) side. The interface then presents the returned privacy policy to the DS, who customizes the policy. When the DS submits the customized policy, representing their informed consent, the policy interface transmits the policy to the privacy manager (DC). The privacy manager (DC) stores the

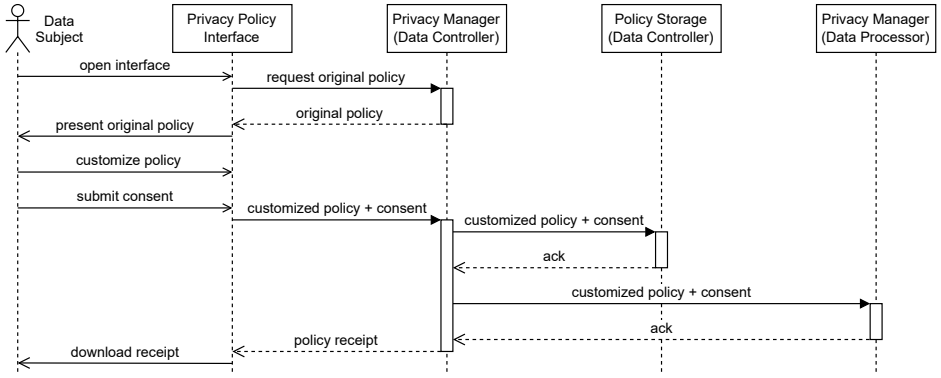

Figure 5: Sequence diagram describing the collection of consent.

policy, and hence the consent, in the Policy Storage (DC). The policy can now be used to prove explicit consent to data protection authorities and is also available for policy enforcement via P2BAC. All data processors, that are involved in processing data for the consented purposes, receive a copy of the customized privacy policy. The PM (DC) transmits the policy to the PM of each individual data processor. To complete the consent collection, the PM (DC) sends a receipt to the privacy policy interface, confirming the successful storing of the policy. The policy interface then downloads the receipt to the device of the DS.

Applied to our example, the procedure can be described as follows. The Data Subject (DS) wants to place an order in our online shop. Therefore, they open the website and are presented with the privacy policy interface. The interface loads the privacy policy of our online shop and allows the DS to customize it. The DS allows processing of their data for shopping and for advertising. The customized policy is transferred to the Privacy Manager (PM) of our online shop and stored in our policy storage. Since the DS allows external processing for advertising and shipping, the policy is forwarded to the PMs of the marketing company and the shipping company. To confirm the customized policy, the PM returns a policy receipt to the policy interface, which the DS can store as a copy of the agreed privacy policy.

Policy Customization. When the Data Subject (DS) wants to customize their privacy policy or wants to withdraw consent for an already accepted purpose, they open the privacy policy interface (cf. Figure 6). The interface loads the stored policy via the Privacy Manager (PM) of the Data Controller (DC). After the DS further customized the policy to their preferences, the interface sends the updated policy, and thus the updated consent, to the PM (DC). The PM (DC) updates the policy in the policy storage, keeping a copy of the previous version of the policy. Each Data Processor (PD), that had previously received a copy of the policy, is also informed about the updated policy. Additionally, any new DPs, which the DS may consent to, will also be informed about the current privacy policy.

The PMs on both sides (DC and DP) inform their corresponding P2BAC about the policy change. This is important when the DS withdraws consent for a purpose, for which data have already been accessed. P2BAC can then prohibit further access to the data for the withdrawn purpose. It can also inform the service, which is processing the data, to stop processing. When the service confirms the successful adaptation to the new policy, P2BAC can confirm adaptation to the PM (DC). Finally, the PM (DC) confirms the policy update to the privacy policy interface, providing a receipt of the newly enforced policy to the DS.

When the Data Subject (DS) decides to withdraw consent for data processing for advertising purposes, after using the online shop for some time, they re-open the privacy policy interface. The interface loads the previously customized policy from our Privacy Manager (PM). After the DS adjusts the policy, withdrawing consent for advertising purposes, the interface submits the new version of the customized policy to our PM. Our PM puts the new policy into the storage and informs the PM of the marketing company about the withdrawn consent. The marketing company stops processing, analogously to the way it is described for internal processing in Figure 6: P2BAC of the marketing company is informed and informs the processes performing the task of interest processing about the withdrawn consent. The process stops processing the data of our DS and confirms this to P2BAC, which

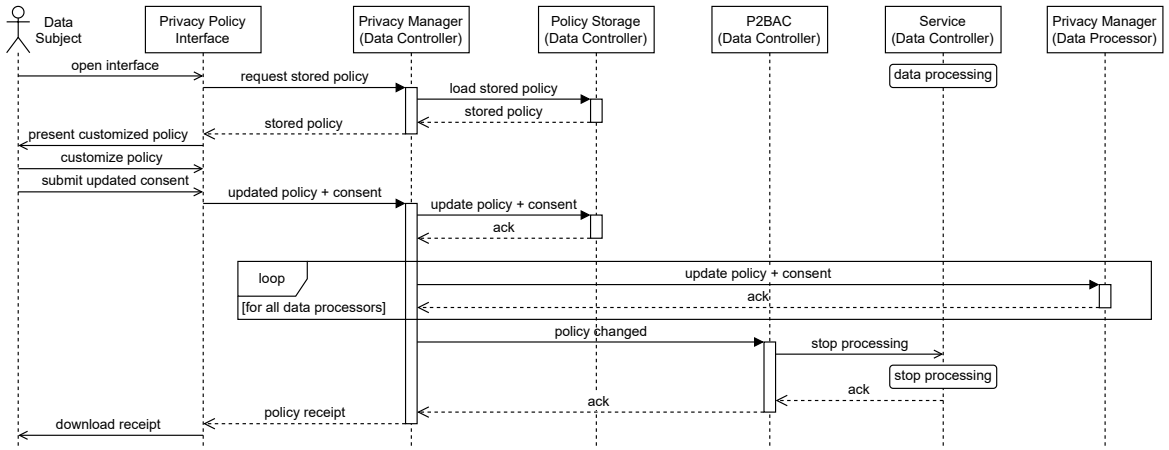

Figure 6: Sequence diagram describing the customization of the policy (partial withdrawal of consent) by the data subject while using the service.

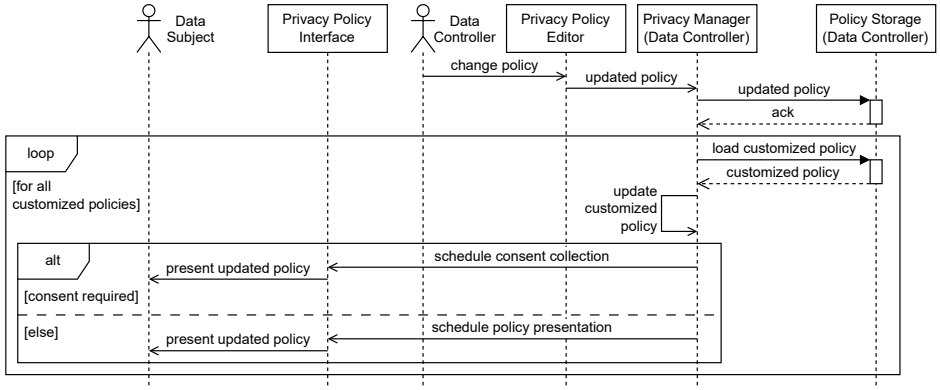

Figure 7: Sequence diagram showing how data subjects are informed when the data controller updates the policy, continued in Figure 8.

confirms policy enforcement to the local PM of the marketing company. Finally, our PM confirms the new policy towards the privacy policy interface, by providing the policy receipt.

Policy Update. A privacy policy update by the Data Controller (DC) is also managed by the privacy policy management. We split this process into two sequence diagrams, Figures 7 and 8.

After the DC makes changes to the policy (cf. Figure 7), using the privacy policy editor, the updated policy is handed to the Privacy Manager (PM) of the DC. The PM (DC) pushes the policy to the policy storage. Next, the PM (DC) checks all the customized policies from the storage and updates them to the new base policy (loop). When the changes require the collection of consent by the Data Subject (DS), the PM (DC) queues the privacy policy interface for the next login of the DS. Else, when no new consent is required, e.g., when corrections are made, or processing is removed from the policy, the privacy policy interface is queued to inform the DS about the updated policy. By default, the updated policies deny access to data, for all new processing purposes that have not been consented, yet.

In Figure 8 we show the rest of the policy update, which happens after the DS is informed about the policy update, or provides consent to the updated policy. The DS can customize the updated policy, provide consent for the new purposes, or dissent them. If no new consent is required, the update process automatically continues after the opt-frame. Regardless of the need for consent described above, the PM (DC) stores the updated policy in the policy storage. Next, the PM (DC) distributes the updated policy to all data processors. Afterward, the PM (DC) informs P2BAC about the updated policy, which informs the service about any adjustments that need to be made concerning data processing.

In our example, we update the privacy policy for our online shop. We add another purpose processing our customers' interests for the creation of customer-specific mail marketing. Our Privacy Manager (PM) saves the new policy in our policy storage. For all our customers, our PM loads their customized policies and compares these with our updated base policy. Since we added a new purpose, requiring consent collection, the PM schedules consent collection by the privacy policy interface. When a customer signs in to our online shop, the privacy policy interface opens

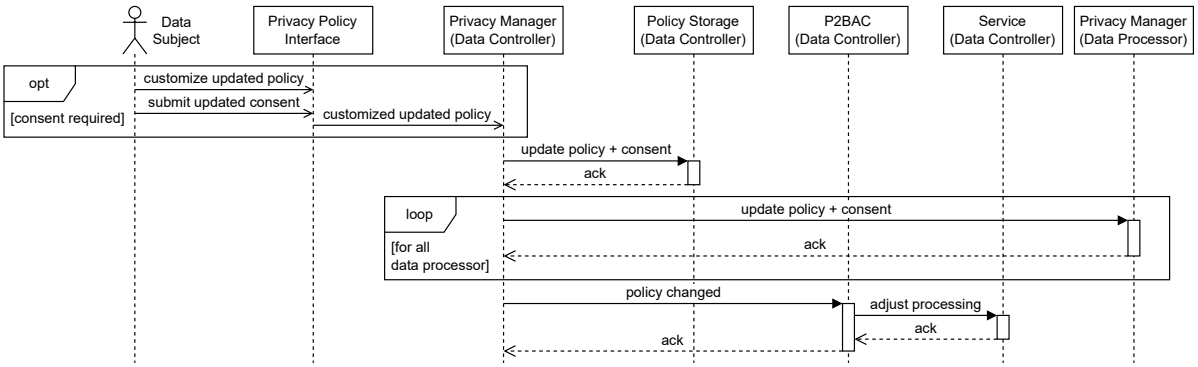

Figure 8: Sequence diagram showing the continuation of the policy update (cf. Figure 7).

automatically. Our customer dissents the new purpose and submits their policy. The policy interface forwards the policy to our PM, which stores the updated policy in storage. Our PM provides the updated policy of the customer to the PMs of the shipping company and the marketing company. The marketing company will only process this customer's data for interest-based advertisement selection, not for mail marketing.

If our policy change would only include corrections or removal of a purpose, we would not need to wait for our customers' consent. The policy interface would be scheduled, to inform our customers about the updated policy. Our PM would store the updated customized policies, which it updated itself according to our base policy (cf. Figure 7), and inform the data processors and P2BAC immediately, without waiting for updated consent.

4.4.2 Data Value Chain

Usage of the Service. The behavior in Figure 9 represents the normal usage of the service. Data, necessary to provide the service, are assumed to be available in the data storage of the Data Controller (DC). In Figure 9, P2BAC provides requested data to the service, when access is granted. This is the case, as the policy enforcement point of P2BAC would provide the requested data, which is not a separate component in our diagram.

When the Data Subject (DS) wants to use the service of the DC, the service requests the necessary data from P2BAC. The policy enforcement system requests the current privacy policy for this DS from policy storage. Based on the loaded policy, P2BAC makes an access decision and either grants access to the data or denies access, preventing data misuse. When access is granted, P2BAC returns the requested data to the service. The service then starts processing the data.

If a data processor is involved in service provision, the service requests the external processing of data from P2BAC. External processing is only triggered, when access to the resulting data is granted. After the access decision, analogously to the one described above, P2BAC requests the data processing at the PM (DC). The PM (DC) forwards the request to the privacy manager of the data processor (DP).

Detailed actions on the DP side are not visualized in Figure 9. However, the PM (DP) triggers the task, which requests necessary data from the corresponding P2BAC. After completing, the task returns any derived data to the PM (DP), similar to the service on the DC's side.

The PM (DP) returns the derived data to the PM (DC), which stores the derived data in storage. Afterward, the PM (DC) informs P2BAC, which then loads the derived data. Finally, P2BAC provides the data to the service. When the service derives new data from the processing of data (by itself or by a data processor), it provides the data to the PM (DC), responsible for storing the data in the database.

When the Data Subject (DS) places an order, the order details are pushed to our Privacy Manager (PM), which stores it in the database. This data col-

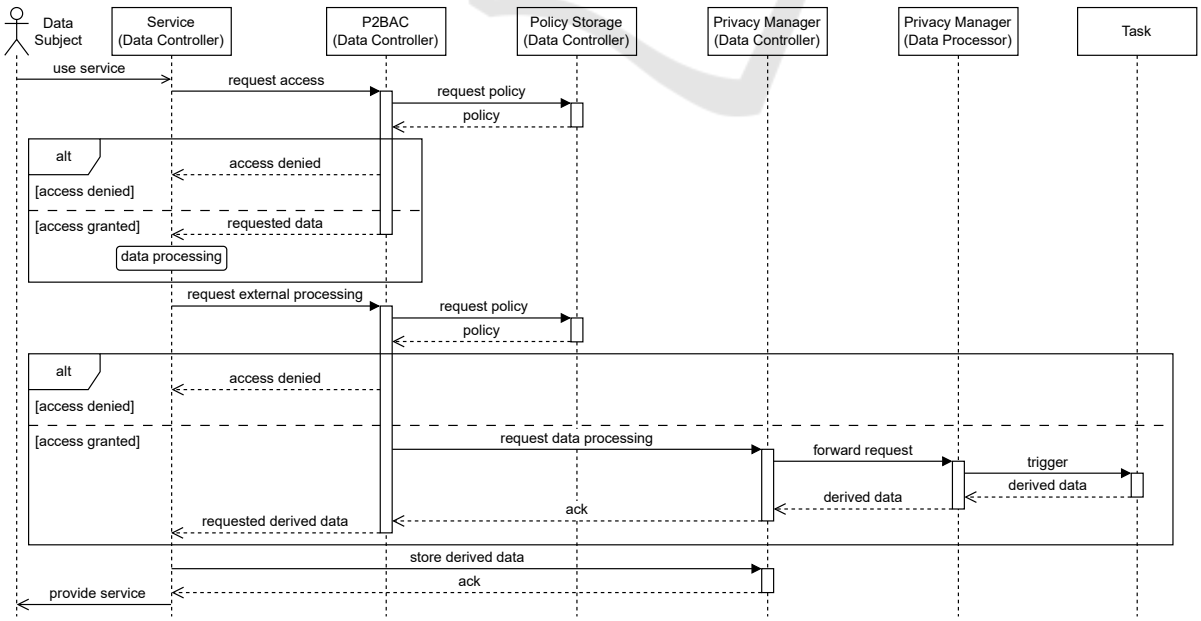

Figure 9: Sequence diagram describing the regular usage of the service.

lection is not shown in Figure 9. Our service needs access to customer data, e.g., the shipping address, to be able to fulfill the order. The shopping process requests this information from our P2BAC instance. Since the DS provided consent for shopping as well as advertising purposes, access is granted and the necessary data are provided to the service. To show interest-based advertisements in the shopping cart, the service requests interest processing by the marketing company. Our P2BAC forwards the request to our PM, which requests the interest processing at the PM of the marketing company. The interests are returned to our PM, which places the information in our database. Finally, P2BAC grants the service access to the information, which then rolls out the best fitting interest-based advertisements.

4.4.3 Transparency

The logs created by the privacy managers support the data controllers in increasing transparency towards their data subjects. We do not visualize interactions with the logs in this paper, but we describe them below.

Data subjects can request a list of data processors from the privacy manager. This request is done via a part of the user interface of the service, e.g., a website. Compared to a state-of-the-art privacy policy, this log provides transparency about the actual data processors of the data, not just about the potential processing of data.

In our online shop example, the data subject will receive a list of all shipping companies, which have transported their orders. Additionally, all marketing companies, that our online shop cooperates with, will be listed.

4.4.4 Accountability

The data transfer logs provide accountability regarding who is processing a data subject's data. This accountability can be used by data protection authorities and police when investigating data breaches. Data protection authorities can also use them to check a controllers GDPR-compliance.

5 RELATED WORK

Enterprise Privacy Authorization Language (EPAL). is a complementary concept to the Platform for Privacy Preferences (P3P) system (Cranor et al., 2006). It was proposed by Ashley, et al., to make privacy enforceable (Ashley et al., 2003). The enforcement concept of EPAL combines two concepts of the PriPoCoG-framework into a single component, while splitting other components apart. Instead of having an access control system and separate policy management, EPAL uses a single enforcement and management component, which accesses a *Privacy Management Server* (Schunter and Ashley, 2002, Figure 1). Logging is not part of EPAL's management, but instead separated into another component.

Our approach benefits from the separation of the enforcement and management, as our Privacy Policy Management (PPM) can take on additional tasks like keeping track of data transfers and providing compliance logs towards data protection authorities. The PPM can also provide the current privacy policy to all data processors. This support along the data value chain is not available in EPAL.

eXtensible Access Control Markup Language (XACML). is a standardized access control policy language and comes with an enforcement and management architecture (OASIS, 2013). XACML uses a Policy Administration Point (PAP) for management of different policies, which retrieves the policies from a Policy Retrieval Point (database or file system).

Compared to our PPM, the PAP is limited to loading and storing the policies, it does not transfer the policies to other XACML systems, nor does it perform logging. Access control logs in XACML are written by the Policy Enforcement Points. How these logs operate for external data processing is unclear.

Sticky Policies. With the sticky policies approach by Pearson and Casassa-Mont "Users can directly control how their data should be processed, handled, and shared by explicitly expressing their preferences and data handling policies." (Pearson and Casassa-Mont, 2011) The user-specific privacy policy is attached directly to the data and transferred as a bundle to all data processors. Based on the attached policy, the data processors know how they have to handle the data.

We make use of the basic concept of transferring a policy together with the data. However, we do not attach the policy directly to the data. Instead, we have a system and database which manage the policies separately from the normal data management of the data handling entities. This allows a decoupling of the type of policy and enforcement, from the general policy management approach. Although we explain our approach using the PriPoCoG-framework it can also be used with different privacy policy systems. Data processors and data controllers may even use different policy languages and systems, as the privacy policy

management could integrate policy converters, translating privacy policies from one policy language to another.

6 CONCLUSION

We presented our Privacy Policy Management (PPM) approach, which integrates into the PriPoCoG-framework (Leicht et al., 2022). The PPM works for the data controller and data processors, and stores and manages the customized policies of the data subjects. It distributes the policies to all data processors, ensuring that every party handling a data subject's data is informed about the agreed upon privacy policy. Updates to the policy by the data controller are compared to already customized policies and data subjects are informed about changes. In case explicit consent is required, the PPM requests this consent from the affected data subjects. When a data subject customizes their privacy policy after submitting initial consent, the PPM takes care of enforcing the withdrawal of consent. The PPM in cooperation with P2BAC (Leicht and Heisel, 2023) ensures that data are only processed according to the customized privacy policy, which is achieved by collecting data via the PPM. Finally, the PPM logs all data processors, keeping track of where data have been transferred.

Looking back at the research questions, stated in Section 1, we conclude that

RQ1. *How can data controllers and data processors be supported, when using customizable privacy policies?*: Our PPM manages customizable privacy policies and data subjects' (partial) consent. The policies are propagated along the data value chain, and all parties involved in data handling and processing work with the latest version of a data subject's privacy policy. Updated policies and consent withdrawal are propagated to all parties that received some data from the data subject.

RQ2. *How can data flows be made more transparent towards the data subjects?*: The logs created by the PPM can be presented to the data subject, so that they can transparently see where their data have been transferred.

Using our PPM data controllers can demonstrate their GDPR-compliance, regarding consent collection, to the data protection authorities. Compared to regular consent mechanisms, we empower the data subjects by allowing them to customize privacy policies; state-of-the-art privacy policies only allow a take-it-or-leave-it approach. This customization is, however, not achieved by the PPM alone, but rather by the complete PriPoCoG-framework, which it integrates into (Leicht et al., 2022).

Although we present our PPM tightly integrated into the PriPoCoG-framework, it can easily be adapted and used with other policy languages and systems. It could for example be integrated into the EPAL or XACML systems.

In the future, we plan to implement a prototype of the proposed PPM and evaluate its applicability. Further work around the PriPoCoG-framework should be put into the privacy policy interface. The policy definition process should also be further improved, to better support data controllers in their work. Improvements towards the data controllers may increase industry acceptance of the framework.

REFERENCES

Ashley, P., Hada, S., Karjoth, G., Powers, C., and Schunter, M. (2003). Enterprise Privacy Authorization Language (EPAL). *IBM Research*, 30:31.

Cranor, L., Dobbs, B., Egelman, S., Hogben, G., Humphrey, J., Langheinrich, M., Marchiori, M., Presler-Marshall, M., Schunter, M., and Wenning, R. (2006). The Platform for Privacy Preferences 1.1 (P3P1. 1) Specification. *W3C Working Group Note*, page 57.

European Parliament and Council of the European Union (2016). Regulation 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, pages 1–88.

Gerl, A. (2020). *Modelling of a Privacy Language and Efficient Policy-Based De-Identification*. PhD thesis, Universität Passau.

Leicht, J. and Heisel, M. (2023). P2BAC: Privacy Policy Based Access Control Using P-LPL. In Mori, P., Lenzini, G., and Furnell, S., editors, *9th International Conference on Information Systems Security and Privacy*, pages 686–697. SciTePress.

Leicht, J. and Heisel, M. (2024). Extending PriPoCoG: A Privacy Policy Editor for GDPR-Compliant Privacy Policies. In *ENASE*, pages 307–318.

Leicht, J., Heisel, M., and Gerl, A. (2022). PriPoCoG: Guiding Policy Authors to Define GDPR-Compliant Privacy Policies. In *Trust, Privacy and Security in Digital Business: 19th International Conference, TrustBus 2022, Vienna, Austria, August 24, 2022, Proceedings*, pages 1–16. Springer.

Leicht, J., Wagner, M., and Heisel, M. (2023). Creating Privacy Policies from Data-Flow Diagrams. In Katsikas, S., Cuppens, F., Cuppens-Boulahia, N., Lambrinoudakis, C., Garcia-Alfaro, J., Navarro-Arribas, G., Nespoli, P., Kalloniatis, C., Mylopoulos, J., Antón,

A., and Gritzalis, S., editors, *Computer Security. ESORICS 2023 International Workshops*, pages 433–453. Springer Nature Switzerland.

OASIS (2013). eXtensible Access Control Markup Language (XACML) version 3.0.

Pearson, S. and Casassa-Mont, M. (2011). Sticky Policies: An Approach for Managing Privacy Across Multiple Parties. *Computer*, 44(9):60–68.

Schunter, M. and Ashley, P. (2002). The Platform for Enterprise Privacy Practices. Technical report, IBM Zurich Research Laboratory. https://www.w3.org/2003/p3pws/pp/ibm3.html.