An Overview of Runtime Data Protection Enforcement Approaches

Laurens Sion
imec-Distrinet, KU Leuven
3001 Leuven
laurens.sion@cs.kuleuven.be

Dimitri Van Landuyt
imec-Distrinet, KU Leuven
3001 Leuven
dimitri.vanlanduyt@cs.kuleuven.be

Wouter Joosen
imec-Distrinet, KU Leuven
3001 Leuven
wouter.joosen@cs.kuleuven.be

***Abstract*—A regulatory framework such as the GDPR succeeds in (i) providing clarity about the nature and the reach of fundamental rights to data privacy and the sovereign role of the data subject, (ii) raising broader awareness of the substantial impact of large-scale, contemporary software-intensive data processing operations on these rights and freedoms, and (iii) creating urgency and imposing gravity, by forcing organizations to take these rights and fundamental principles seriously in a proactive manner.**

However, regulatory frameworks lack clarity on how these concerns are to be enacted. For example, guidance is lacking on how software should be constructed to consider these data protection principles by design and by default. In this paper, we argue how the direct translation of the GDPR data protection principles into design or code falls short in the context of contemporary software systems, which are both more dynamic and nature and rely on an increasing number of complex inter-organizational collaborations. This means that in such a system, data protection decisions cannot be 'hard-coded' but will have to be decided at run time. In addition, we provide an overview of promising existing approaches that contribute to the accomplishment of these fundamental data protection principles at run time.

***Index Terms*—privacy by design, data protection, GDPR, compliance, legal**

1. Introduction

The General Data Protection Regulation (GDPR) [1] has introduced Data Protection by Design and by Default as key principles to incorporate in data processing operations. These principles have far-reaching consequences on the design and development of software systems, as they impose a wide range of obligations on the processing of personal data, the purposes for which it can be processed, the handling of data subject requests, etc.

Common interpretations of these data protection principles focus on a translation of the encompassed obligations into requirements to implement in code or design. Such approaches meet these principles by directly translating them into code and implementation artifacts that directly enforce them in the resulting software product, and many proposals in the literature such as UML extensions and requirements approaches follow this path [2]–[13].

These types of approaches effectively hardcode the data protection decisions up front in the design or implementation of the software system. However, many of these decisions can only be made at run time and, hence, require solutions that, while instantiated in the design or implementation, only make the actual data protection decisions at run time. For example, while the implementation of a software product could encode that any personal information entering the system requires consent, a more natural approach is to decide at run time which types of data processing operations can occur, taking into account the consent decisions of the user more dynamically. Furthermore, as data subjects can revoke consent at any time, the system cannot rely on it up front. Instead, the system will need to perform a check at run time to ensure consent has not been revoked.

Furthermore, there are numerous cases where the requirements are inherently dynamic. For example, an appropriate implementation of the principle of storage limitation may rely on flexible criteria to be met in the future to determine if the data should no longer be kept. Another example is when integrating with third parties, for providing services to some users, and, hence, requires an additional notice to those data subjects when their information would be communicated to those third parties. Comprehensively addressing the GDPR's obligations requires approaches that can make decisions at run time to enforce the relevant data protection principles.

In this paper, we: (i) argue why the dynamic and complex nature of contemporary development practices and software services can make exclusively relying on static up front enforcement of the GDPR obligations suboptimal; (ii) provide an overview of existing approaches and how these approaches support and enable our vision. (iii) discuss the advantages and disadvantages of enforcing data protection decisions at run time instead of up front.

This paper is structured as follows. Section 2 first provides some background on the GDPR's data protection principles and the state of the art in compliance engineering. Section 3 motivates the paper from an example application. Section 4 provides an overview of the different approaches, the GDPR principles to which they contribute, and whether they encode enforcement decisions up front or at run time. Section 5 discusses the advantages and disadvantages of addressing data protection at run time. Finally, Section 6 concludes the paper.

2. Background

Section 2.1 first outlines the data protection principles of the GDPR. Next, Section 2.2 presents the current state of the art in approaches aimed at attaining compliance

https://doi.org/10.1109/EuroSPW54576.2021.00044

by design in the early stages of software development (requirements, architecture, and design).

2.1. Data Protection Principles

This section briefly presents the data protection principles as outlined in Art. 5 of the GDPR [1]. For a more elaborate description of these principles and their interpretations, we refer the reader to the GDPR [1] and the relevant guidelines [14]–[17].

Lawfulness, fairness, transparency Ensuring valid lawful ground, such as consent, is obtained for the processing, including the relevant additional constraints. For example, consent has to be specific, informed, voluntary, etc. Furthermore, for transparency, the data subject must be adequately informed about the data processing operations.

Purpose limitation Specifying a specific purpose for the processing and ensuring any further processing of personal data is compatible with the purpose specified when collecting the personal data.

Data minimization Only collect and process the minimal amount of personal data that is necessary for the processing purpose.

Accuracy Ensure that personal data are accurate and kept up to date. Any inaccurate personal data can be corrected or removed.

Storage limitation Personal data is only kept in an identifiable form as long as is necessary for the purpose of the processing.

Integrity/confidentiality Personal data needs to be processed in manner to ensure integrity and confidentiality of the personal data.

Accountability The controller is responsible for the data processing and able to demonstrate compliance with the provisions of the GDPR.

2.2. State of the art in compliance engineering

We highlight a number of approaches that take place in the early stages of development and lead to a static, hard-coded treatment of data protection principles.

2.2.1. Translation of legal obligations to requirements. These approaches rely on the analysis of legal texts to extract requirements for a software system that have to be realized. These requirements are then integrated in the requirement corpus of the system and addressed in subsequent development steps.

- Models and languages for capturing and translating legal obligations into requirements [18]–[20]
- Legal texts in requirements engineering [21]

2.2.2. Verifying compliance of a system against the GDPR. Data protection impact assessment (DPIA) approaches [4], [22], [23] involve extensive reasoning about legal obligations at the basis of legal abstractions, typically encoded in models. These models can subsequently be analyzed to assess and identify problematic data processing operations. Apart from providing support towards organizational and legal mitigations, they as such contribute to the identification of system requirements.

These analysis approaches commonly rely upon model-based representations of the data protection principles, either as separate modeling abstractions [2]–[4], [12], [13], or as extensions, such as UML profiles, on existing modeling languages [5]–[11].

2.2.3. Enforcement of legal requirements in software design. Not starting from the GDPR, but from system descriptions instead are the approaches to elicit legal requirements. These approaches rely on some knowledge base or repository of problematic patterns in a software system that can pose a privacy problem (e.g., privacy patterns). The systematic analysis of a system model can subsequently assist in identifying all these problematic situations and suggest appropriate countermeasures to prevent them from occurring.

Because of the limited information in such system description models to support the analysis for privacy problems, several proposals in the literature introduce extensions to these models with additional privacy information in support of more extensive analyses [24]–[26].

Privacy engineering approaches [10] advocate addressing these requirements in the development life cycle, by adopting privacy design strategies [27], [28], augmenting design models [29], considering data protection principles, such as data minimization [30], during development, and adopting and implementing privacy patterns [31] and privacy-enhancing technologies (PETs) [32], [33].

3. Motivation

The software industry has evolved from a product-oriented into a service-based economy, and, as a consequence, contemporary software systems are dynamic, complex, and distributed software systems. As such, these systems dynamically engage in inter-organizational data transfers, and these come with important and relevant data protection implications. This section introduces an example of such an application and motivates the importance of enacting the data protection principles at run time.

This example is representative of the operational reality of a Belgian Software-as-a-Service (SaaS) B2B provider of a document generation and delivery service to companies (customer organizations). As depicted in Figure 1, this service accepts raw data batches from customer organizations, uses this raw data to generate PDF documents (e.g., invoices and pay slips) in an automated fashion, and finally, initiates the delivery of these documents to the intended recipients (e.g., customers or employees) via a plethora of delivery channels: integrated into online banking applications (for the delivery of invoices), via postal mail, email, etc.

Organizations that rely on the service only need to provide a template and the appropriate raw data, which are then used by the service to generate the PDF documents and deliver them to the end users by a certain deadline.

From the point of view of the SaaS provider, and given the flexibility in scheduling and processing these large document generation jobs, considerable cost savings can be obtained by frequently re-evaluating the pricing and performance properties of different cloud providers in order to choose the most optimal solution. As such, the actual document generation processes are performed at various cloud platforms (IaaS or PaaS providers such

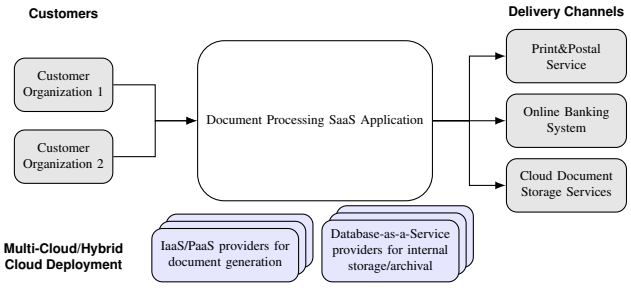

Figure 1. Graphical depiction of the document processing service in its context of a wider inter-organizational and dynamic ecosystem of software-based services

as Amazon AWS or Google AppEngine) in a dynamic multi-cloud manner. In a similar manner, data storage and persistence is accomplished through a mix of on-premise and cloud storage resources.

It is clear that the data processed by the document processing service is to be considered personal information, as it contains identifying information of recipients (for example, required for addressing), but also financial information (in pay slips) and information about activities and services rendered (in invoices and pay slips).

As a consequence, the main data controller (the customer organization that outsources the document generation and delivery) will be required to implement the data protection principles, for example to ensure that access to data is controlled, that no more data is collected and disclosed than necessary for the purpose of the activities (*purpose limitation*), that the data is not kept longer than absolutely necessary (*storage limitation*), etc.

However, the enactment and implementation of these principles purely in a constructive software design context, e.g., during the implementation of the document processing service, will not be sufficient, as the document processing service provider itself, at construction time, will be unaware of the different delivery channels and the different cloud providers it will use to accomplish the overall service. Indeed, support for different delivery channels may be added later, and customer organizations may prefer certain channels to deliver the documents. Furthermore, even changes external to the system such as the invalidation of the EU-US Privacy Shield [34] can have important implications in terms of which third parties this data can be shared with. In this case, their geographic location and the existence of other safeguards needs to be considered.

The above implies that the enactment of these principles partially will need to shift away from the development to the operational context of the system so that decisions regarding these principles can be made with recent and up-to-date information. For example, we can envision that the list of concerns typically taken into account and established in contract during service negotiation with the various involved cloud providers (performance, availability, pricing) will have to be extended with clauses about data protection. The selection of a suitable service provider will be required to also take into account, for example, the geographic location of the cloud provider, the data protection control mechanisms it offers, the guarantees it provides about operational security and data retention.

The next section will look into a wide range of data protection approaches to assess to which extent they can be applied in such a runtime context.

4. Overview of data protection approaches

A wide range of approaches can be applied in support of the GDPR's data protection principles. First, Section 4.1 revisits the data protection principles from Section 2.1 to describe the interpretation of meeting these principles at design or runtime. Next, Section 4.2 discusses the taxonomy of the categories of runtime approaches and to which data protection principles they contribute.

4.1. Design and runtime data protection principles

For each of the data protection principles of Section 2.1, this section discusses the distinction between addressing them at design or at run time.

Lawfulness, fairness, transparency

Design This entails determining the lawful grounds and transparency obligations upfront and in a static manner. Any changes such as revoked consent, or differences in the processing of personal data have to either be considered and implemented in the system upfront or require manual intervention of the organization afterwards.

Runtime The runtime approach entails dynamically considering the preferences of individual users and, for transparency, dynamically selecting which information needs to be communicated to specific data subjects.

Purpose limitation

Design The purposes of the different processing operations are determined and fixed upfront. It is not possible to add or change purposes because it is not possible to determine if these are compatible.

Runtime This system is able to determine at run time whether the use of certain elements of personal data for a specific purpose are compatible with the purpose of collection and is able to dynamically select only those data elements for which purpose compatibility is met.

Data minimization

Design This involves determining the minimal amount of data that is needed for the processing operations. Any differences between types of users have to be considered and implemented upfront.

Runtime The system is able to determine the minimal amount of data that is used for each individual data subject at run time depending on the use of different functionalities of the system by the data subject.

Accuracy

Design The design time consideration of accuracy either requires a direct translation to a functional requirement in the system that enables data subjects to alter their personal information or offloads these requests to the organization to manually resolve these issues.

Runtime For some types of data the system could determine at run time that they are no longer correct (e.g., email messages are undeliverable) and correct these data entries accordingly.

Storage limitation

Design The time period for how long data needs to be stored is decided upfront; fixed data retention policies are implemented.

Runtime The duration of storage can be determined dynamically for each data subject and data item, and the system can respond by automatically removing data that should no longer be stored. A dynamic and individual retention policy is supported in a flexible data management system that decides on a per-data-subject basis.

Integrity/confidentiality

Design The selection and integration of fixed security countermeasures, such as encryption, to protect the confidentiality of data.

Runtime The system can apply certain countermeasures automatically. For example, access control mechanisms determine at runtime if some user should be able to access certain personal information, and this access control logic can change as access control policies are modified, or at the basis of more dynamic conditions (e.g., only when there is uncertainty about the identity of a user, a multi-factor authentication policy can be dynamically enabled for stronger guarantees about the identity).

Accountability

Design Design-time accountability approaches focus on the ability demonstrate compliance artifacts from a design perspective, e.g., static privacy policies or data protection impact assessment reports. Accountability at design time means that the involved organization can rely upon sufficient design and process documentation to showcase it has taken a data protection-by design approach.

Runtime These ensure that the system can continuously demonstrate its compliance through, for example, secure logs and auditing. Run-time accountability refers to the capability of a system to keep track of individual interactions, and has the ability to provide evidence (e.g., from audit logs) that it has behaved in accordance with data protection principles.

4.2. Categories of approaches

This section describes the result of an exploration of the different approaches that can be applied to address the GDPR's data protection principles at runtime. The approaches are categorized in a bottom-up fashion by grouping similar types of approaches together based on shared mechanisms (e.g., policies, monitoring), or elements they affect (e.g., data or user interaction). For each approach, the coverage of the different data protection principles is described (both design and runtime) using the interpretations outlined above. Table 1 shows the resulting overview of these technologies and the principles to which they contribute. Each of the categories are discussed in more detail below.

4.2.1. User Preferences. These solutions are applied in the system at the point of interaction with the data subjects. They primarily focus on: These technologies focus on the interaction with the user, considering: (i) lawfulness, by, for example, ensuring consent has been obtained from the data subject, (ii) transparency, by clarifying the impact of sharing certain data to the data subject, and (iii) data minimization, by informing or nudging the data subject to share less personal information.

4.2.2. Data management. Data management techniques focus on the manipulation of the collected or processed data to meet the principles. These techniques primarily contribute to data minimization by: (i) limiting the collected data upfront, (ii) processing the data to remove identifying information, and (iii) limiting the information that is revealed when using the data. They also contribute towards (i) storage limitation, by supporting the deletion of personal information that is nog longer needed, and (ii) confidentiality and integrity, by encrypted the transfer and storage of personal information.

4.2.3. Policies. Policy technologies cover a wider range of applications, ranging from end-user privacy policies, such as P3P [48] to application-layer purpose-based access control systems [54]. Policy technologies address the following principles: (i) lawfulness, fairness, and transparency, by communicating privacy policies to end-users and enforcing processing is compliant with end-user restrictions (e.g., consent); (ii) purpose limitation, by enforcing any further processing is compatible with the purpose of collection by restricting access to compatible processing operations; (iii) data minimization, by limiting the information that has to be provided by data subjects for access control, (iv) storage limitation, by automating the deletion of data when certain criteria are met, (v) integrity and confidentiality, by enforcing access control when personal data is accessed in the system.

4.2.4. Monitoring. Monitoring technologies primarily support the transparency principle by tracking the processing operations and making this information available to data subjects. Monitoring can also be used as a mechanism to detect violations against the other GDPR principles at runtime. One technology explicitly designed for the purpose is the minimization monitoring [57], although other monitoring mechanisms as part of, for example a purpose-based access control system, could analogously be used to detect violations against other principles such as purpose limitation or confidentiality.

5. Discussion on the advantages and disadvantages of run-time enactment of data protection

The previous section provided an overview of how key data protection principles can be enacted by a system during its run time, as opposed (and complementary) to the more static, by-design implementation of these data protection principles, e.g., during requirements elicitation, architecture design or implementation. This section discusses the advantages and disadvantages of run-time decisions to enforce data protection principles in software systems. The discussion starts with the problems of a design-time encoding of data protection decisions and how the run-time enforcement can resolve these problems. Next,

TABLE 1. OVERVIEW OF ENABLING TECHNOLOGIES AND TO WHICH DATA PROTECTION PRINCIPLES THEY CONTRIBUTE.

Category / **Technology**	*Lawfulness, fairness, and transparency*		*Purpose limitation*		*Data minimization*		*Accuracy*		*Storage limitation*		*Integrity & confidentiality*		*Accountability*	
	D	R	D	R	D	R	D	R	D	R	D	R	D	R
User Preferences														
Informed privacy settings [35]	○	●	○	○	○	◐	○	○	○	○	○	○	○	○
Consent Management Platforms [36]–[38]	○	●	○	○	○	○	○	○	○	○	○	○	○	●
Privacy nudges [39]–[42]	○	◐	○	○	○	◐	○	○	○	○	○	○	○	○
Data Management														
Data minimization [24]	○	○	○	○	●	○	○	○	○	○	○	○	○	○
De-identification [43]	○	○	○	○	●	○	○	○	○	○	○	○	○	○
Differential privacy [44]	○	○	○	○	○	●	○	○	○	○	○	○	○	○
Local differential privacy [45]	○	○	○	○	○	●	○	○	○	○	○	○	○	○
Compliant database [46]	○	○	○	○	○	○	○	○	○	●	●	○	○	●
Policies														
P3P [47], [48]	●	●	◐	◐	○	○	○	○	○	○	○	○	◐	○
Enterprise privacy policies [49]	○	●	○	●	○	◐	○	○	○	●	○	◐	○	○
Sticky policies [50]–[52]	○	◐	○	●	○	○	○	○	○	○	○	◐	○	○
Access Control (Data User) [53], [54]	○	○	○	●	○	○	○	○	○	○	○	◐	○	○
Access Control (Data Subject) [55]	○	●	○	○	○	◐	○	○	○	○	○	○	○	○
Privacy API [56]	○	●	○	●	○	○	○	○	○	○	○	○	○	○
Monitoring														
Minimization monitoring [57]	○	○	○	○	○	●	○	○	○	○	○	○	○	○
Logging [58]–[60]	○	●	○	○	○	○	○	○	○	○	○	○	○	●
Workflow auditing [61]	○	●	○	○	○	○	○	○	○	○	○	○	○	●

Legend: ***D*** *design time decision,* ***R*** *runtime decision,* ● *primary principle the technology contributes to (not necessarily complete so other technologies can still be required),* ◐ *other principles the technology can contribute to.*

some of the disadvantages and complications of run-time enforcement are discussed.

5.1. Advantages

One of the primary advantages of run-time decisions is that it allows addressing the data protection requirements that can simply not be met by making the decisions upfront. An example of this problem is when the processing of personal data relies on consent of the data subject. While it is possible to assume upfront that such consent has been provided, the fact that a data subject may revoke consent at any time may cause such processing to no longer be lawful. Systems in which consent is checked at run time before any processing occurs can readily take this change in consent into account and adjust the processing of personal information accordingly.

Another advantage of run-time decision-making is that aligns better with contemporary software services which frequently rely on third party service providers for infrastructure, storage, or computing. Given the dynamic pricing of these services, considerable cost savings can be obtained by automating the selection of these services based on their dynamic prices. Such a dynamic context implies changes to the running system with important data protection implications. For example, the system could determine the appropriate privacy notices at runtime and customize them to the actual situation instead of a blanket statement describing the worst case of relying on all the different service providers. In addition to the changes in service providers, modern cloud deployments also introduce issues of data locality as personal data can not readily be transferred across borders. Depending on the location of the organization and the data subject, additional restrictions may apply to these transfers which have to be taken into account by the system.

Furthermore, by addressing data protection principles at runtime, several obligations from the GDPR can be automated. There are several examples of this: (i) the removal of information that is no longer need (*data minimization*); (ii) checking whether processing for a certain purpose is permissible (*purpose limitation*); (iii) the detection of inaccurate information such as email address (*accuracy*); and (iv) continuously demonstrating compliance through auditing and logging, and making this information available to data subjects (*accountability*).

A final advantage is that addressing the principles at runtime enables the system to adapt to meet the restrictions and obligations for each individual subject. Instead of designing the system for different types or categories of users, it can adapt the processing to the preferences for every individual user.

5.2. Disadvantages

There are also a number of disadvantages to the runtime approaches. Three are discussed in more detail below.

One disadvantage is the limited coverage of the data protection principles. As Table 1 illustrates, not all principles have extensive support available to address them at runtime. Furthermore, even for the supported principles, the coverage of the runtime technologies focusses on specific aspects such as, for example, consent, but addressing consent at runtime is insufficient for meeting all the obligations of lawfulness, fairness, and transparency.

Another disadvantage is the need for monitoring. As the decisions are no longer made by a human upfront, the application of the runtime techniques requires the instantiation of monitoring and logging mechanisms as a safeguard to identify and remediate in case suboptimal or incorrect decisions are made. Some techniques already directly provide such functionality (e.g., auditing to demonstrate compliance for *accountability*).

Addressing data protection principles at runtime can introduce additional overhead and complexity in the system. For example, additional logging functionality in the database system for accountability introduces a performance cost [46], the runtime checking of purpose compatibility requires detailed encodings of processing purposes (for example, from the Data Privacy Vocabulary [62]) to able to determine purpose compatibility at runtime.

5.3. Final remarks

The problems with the static upfront encoding of data protection concerns were identified by Koops and Leenes [63] in an earlier draft of the GDPR. They extensively discussed the problems with encoding the GDPR's privacy requirements in software, highlighting similar problems with the encoding of requirements. Their example on "*data removal seven years after contract expiration*", illustrates well how certain types of requirements can only be met at runtime and cannot be 'statically hardcoded'.

They also identify the dynamic and fluid nature of many legal norms. While simple rules can be readily translated into a software implementation, more complex, open, contextual rules are problematic. This issue reoccurs in the context of purpose limitation in which the implementation has to make the determination whether certain purposes are compatible. While the enforcement at runtime does not solve these problems, the reliance on updatable data structures of purposes, such as, for example, the Data Privacy Vocabulary [62], can assist in assessing the compatibility of encountered purposes and enable the system to adapt to changing legal contexts without requiring costly reimplementations.

6. Conclusion

The motivation for this paper stems from the realistic example of a modern Software-as-a-Service (SaaS) document generation and delivery service. This application illustrates the problems of any approach that involves directly translating the GDPR data protection principles into static implementation (requirements, design, code). The main cause is that for many contemporary systems, important decisions are still to be made after initial development, at run time. In the case of the document generation and delivery service, it is part of a wider inter-organizational ecosystem of software-based services, and not all parties and services are known at design time, as the system is capable of autonomously selecting, for example, a delivery service at the basis of dynamic criteria such as cost. When making such dynamic selections, it is necessary to consider data protection concerns at run time, exactly because the information to make such decisions is not

available up front. This was illustrated in the motivating example, in which the dynamic selection of delivery, cloud storage, and processing services makes it impossible to make these decisions up front.

With this in mind, we have conducted an in-depth exploration and comparison of the differences between approaches that enact the core data protection principles at design- and at runtime, showing ultimately that these can complement each other. Subsequently, based on an exploration of the literature and the state of the art, we present a comprehensive overview of existing and promising approaches to enact data protection principles at runtime.

This overview allows us to make a number of observations and conclusions: (i) no single approach provides full coverage of the different data protection principles; (ii) the first three principles (*lawfulness, fairness, and transparency*, *purpose limitation*, and *data minimization*) are covered most frequently; and (iii) the coverage of an approach for one principle is frequently limited to a single aspect (e.g., focus on consent for the lawfulness, fairness and transparency principle).

As such, the presented overview provides pointers to promising and more dynamic compliance engineering approaches, that require further investigation and extension, both from a legal perspective as from the perspective of engineering complex adaptive systems that allow and engage in significant non-trivial runtime decision-making.

The focus of this paper has been on the data protection principles outlines in the European GDPR. However, given the strong requirements it imposes and the inspiration it provides for legislative initiatives in other countries, we argue that the identified approaches to meet these principles will also be applicable and relevant in the context of other regulatory frameworks.

Acknowledgements. This research is partially funded by the Research Fund KU Leuven, the PRiSE C2 research project, and the Cybersecurity Initiative Flanders.

References

[1] European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC," *Official Journal of the European Union*, vol. 59, no. L 119, pp. 1–88, May 2016.

[2] J. Tom, E. Sing, and R. Matulevičius, "Conceptual representation of the GDPR: Model and application directions," in *Perspectives in Business Informatics Research*, J. Zdravkovic, J. Grabis, S. Nurcan, and J. Stirna, Eds. Cham: Springer International Publishing, 2018.

[3] D. Torre, G. Soltana, M. Sabetzadeh, L. C. Briand, Y. Auffinger, and P. Goes, "Using models to enable compliance checking against the GDPR: An experience report," *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pp. 1–11, 2019.

[4] L. Sion, P. Dewitte, D. Van Landuyt, K. Wuyts, P. Valcke, and W. Joosen, "DPMF: A modeling framework for data protection by design," *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, vol. 15, no. Special Issue on Privacy in IS Design, pp. 10:1–53, 2020.

[5] A. S. Ahmadian, D. Strüber, V. Riediger, and J. Jürjens, "Model-based privacy analysis in industrial ecosystems," in *Modelling Foundations and Applications*, A. Anjorin and H. Espinoza, Eds. Cham: Springer International Publishing, 2017, pp. 215–231.

[6] M. Alshammari and A. Simpson, "A model-based approach to support privacy compliance," *Information & Computer Security*, vol. 26, no. 4, pp. 437–453, 2018.

[7] ——, "A UML profile for privacy-aware data lifecycle models," in *Computer Security*, S. K. Katsikas, F. Cuppens, N. Cuppens, C. Lambrinoudakis, C. Kalloniatis, J. Mylopoulos, A. Antón, and S. Gritzalis, Eds. Cham: Springer International Publishing, 2018.

[8] T. Basso, L. Montecchi, R. Moraes, M. Jino, and A. Bondavalli, "Towards a UML profile for privacy-aware applications," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 371–378.

[9] D. N. Jutla, P. Bodorik, and S. Ali, "Engineering privacy for big data apps with the unified modeling language," in *2013 IEEE International Congress on Big Data*, 2013, pp. 38–45.

[10] Y. Martin and A. Kung, "Methods and tools for GDPR compliance through privacy and data protection engineering," in *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, Apr. 2018, pp. 108–111.

[11] E. Mougiakou and M. Virvou, "Based on GDPR privacy in UML: Case of e-learning program," in *2017 8th International Conference on Information, Intelligence, Systems Applications (IISA)*, 2017.

[12] S. Sen, S. Guha, A. Datta, S. K. Rajamani, J. Tsai, and J. M. Wing, "Bootstrapping privacy compliance in big data systems," in *2014 IEEE Symposium on Security and Privacy*, May 2014, pp. 327–342.

[13] R. Slavin, X. Wang, M. B. Hosseini, J. Hester, R. Krishnan, J. Bhatia, T. D. Breaux, and J. Niu, "Toward a framework for detecting privacy policy violations in android application code," in *Proceedings of the 38th International Conference on Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2016.

[14] Article 29 Working Party, "Guidelines on transparency under Regulation 2016/679," 2018.

[15] ——, "Opinion 06/2014 on the notion of legitimate interests of the data controller under article 7 of directive 95/46/EC (WP217)," 2014.

[16] ——, "Guidelines on consent under the regulation 2016/679 (WP259 rev.01)," 2018.

[17] ——, "Opinion 03/2013 on purpose limitation) (WP203)," 2013.

[18] T. D. Breaux, M. Vail, W., and A. I. Anton, "Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations," in *14th IEEE International Requirements Engineering Conference (RE'06)*, Sep. 2006, pp. 49–58.

[19] T. D. Breaux and A. I. Antón, "Analyzing regulatory rules for privacy and security requirements," *IEEE Transactions on Software Engineering*, vol. 34, no. 1, pp. 5–20, Jan. 2008.

[20] A. Siena, A. Perini, A. Susi, and J. Mylopoulos, "A meta-model for modelling law-compliant requirements," in *2009 Second International Workshop on Requirements Engineering and Law*, Sep. 2009, pp. 45–51.

[21] P. N. Otto and A. I. Antón, "Managing legal texts in requirements engineering," in *Design Requirements Engineering: A Ten-Year Perspective*, K. Lyytinen, P. Loucopoulos, J. Mylopoulos, and B. Robinson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 374–393.

[22] F. Ferra, I. Wagner, E. Boiten, L. Hadlington, I. Psychoula, and R. Snape, "Challenges in assessing privacy impact: Tales from the front lines," *Security and Privacy*, vol. 3, no. 2, p. e101, 2020.

[23] R. Gellert, "Understanding the notion of risk in the General Data Protection Regulation," *Computer Law & Security Review*, vol. 34, no. 2, pp. 279–288, Apr. 2018.

[24] T. Antignac, D. Sands, and G. Schneider, "Data minimisation: A language-based approach," in *ICT Systems Security and Privacy Protection*, S. De Capitani di Vimercati and F. Martinelli, Eds. Cham: Springer International Publishing, 2017, pp. 442–456.

[25] G. Danezis, J. Domingo-Ferrer, M. Hansen, J.-H. Hoepman, D. L. Metayer, R. Tirtea, and S. Schiffner, "Privacy and data protection by design - from policy to engineering," Tech. Rep., 2014.

[26] A. S. Ahmadian, D. Strüber, V. Riediger, and J. Jürjens, "Supporting privacy impact assessment by model-based privacy analysis," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, pp. 1467–1474.

[27] J.-H. Hoepman, "Privacy design strategies (the little blue book)," 2018.

[28] M. Colesky, J. C. Caiza, J. M. Del Alamo, J.-H. Hoepman, and Y.-S. Martín, "A system of privacy patterns for user control," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, pp. 1150–1156.

[29] F. Burmeister, P. Drews, and I. Schirmer, "A privacy-driven enterprise architecture meta-model for supporting compliance with the general data protection regulation," in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.

[30] A. Senarath and N. A. G. Arachchilage, "A data minimization model for embedding privacy into software systems," *Computers & Security*, vol. 87, Nov. 2019.

[31] L. Compagna, P. El Khoury, A. Krausová, F. Massacci, and N. Zannone, "How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns," *Artificial Intelligence and Law*, vol. 17, no. 1, pp. 1–30, Mar. 2009.

[32] R. Meis and M. Heisel, "Pattern-based representation of privacy enhancing technologies as early aspects," in *International Conference on Trust and Privacy in Digital Business*. Springer, 2017.

[33] J. Curzon, A. Almehmadi, and K. El-Khatib, "A survey of privacy enhancing technologies for smart cities," *Pervasive and Mobile Computing*, vol. 55, pp. 76–95, 2019.

[34] Court of Justice of the European Union, "C-311/18 Data Protection Commissioner v Facebook Ireland Limited and Maximillian Schrems," 2020, ECLI:EU:C:2020:559.

[35] S. J. De and D. Le Métayer, "Privacy risk analysis to enable informed privacy settings," in *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2018, pp. 95–102.

[36] OneTrust, "Consent management platform (cmp)," https://www.preferencechoice.com/consent-management-platform/, 2021.

[37] TrustArc, "Cookie consent manager," https://trustarc.com/cookie-consent-manager/, 2021.

[38] Cookiebot, "Cookiebot cmp," https://www.cookiebot.com/en/, 2021.

[39] A. Acquisti, I. Adjerid, R. Balebako, L. Brandimarte, L. F. Cranor, S. Komanduri, P. G. Leon, N. Sadeh, F. Schaub, M. Sleeper, Y. Wang, and S. Wilson, "Nudges for privacy and security: Understanding and assisting users' choices online," *ACM Computing Surveys*, vol. 50, no. 3, Aug. 2017.

[40] H. Almuhimedi, F. Schaub, N. Sadeh, I. Adjerid, A. Acquisti, J. Gluck, L. F. Cranor, and Y. Agarwal, "Your location has been shared 5,398 times! A field study on mobile app privacy nudging," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2015, pp. 787–796.

[41] L. Warberg, A. Acquisti, and D. Sicker, "Can privacy nudges be tailored to individuals' decision making and personality traits?" in *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society*, ser. WPES'19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 175–197.

[42] Y. Wang, P. G. Leon, A. Acquisti, L. F. Cranor, A. Forget, and N. Sadeh, "A field trial of privacy nudges for facebook," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 2367–2376.

[43] S. L. Garfinkel, "De-identification of personal information," *National institute of standards and technology*, 2015.

[44] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–19.

[45] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 2013, pp. 429–438.

[46] A. Shah, V. Banakar, S. Shastri, M. Wasserman, and V. Chidambaram, "Analyzing the impact of GDPR on storage systems," in *11th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 19)*, 2019.

[47] J. Reagle and L. F. Cranor, "The platform for privacy preferences," *Communications of the ACM*, vol. 42, no. 2, pp. 48–55, Feb. 1999.

[48] L. F. Cranor, "P3P: Making privacy policies more useful," *IEEE Security and Privacy*, vol. 1, no. 6, pp. 50–55, Nov. 2003.

[49] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, "Enterprise privacy authorization language (EPAL)," *IBM Research*, vol. 30, p. 31, 2003.

[50] S. Pearson and M. Casassa-Mont, "Sticky policies: An approach for managing privacy across multiple parties," *Computer*, vol. 44, no. 9, pp. 60–68, Sep. 2011.

[51] S. Sicari, A. Rizzardi, D. Miorandi, and A. Coen-Porisini, "Security towards the edge: Sticky policy enforcement for networked smart objects," *Information Systems*, vol. 71, pp. 78–89, Nov. 2017, https://www.sciencedirect.com/science/article/pii/S0306437917303770.

[52] D. Miorandi, A. Rizzardi, S. Sicari, and A. Coen-Porisini, "Sticky policies: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 12, pp. 2481–2499, 2020.

[53] J.-W. Byun, E. Bertino, and N. Li, "Purpose based access control of complex data for privacy protection," in *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies*, ser. SACMAT '05. New York, NY, USA: Association for Computing Machinery, 2005, pp. 102–110.

[54] F. Pallas, M.-R. Ulbricht, S. Tai, T. Peikert, M. Reppenhagen, D. Wenzel, P. Wille, and K. Wolf, "Towards application-layer purpose-based access control," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, ser. SAC '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1288–1296, https://doi.org/10.1145/3341105.3375764.

[55] C. A. Ardagna, S. De Capitani di Vimercati, S. Paraboschi, E. Pedrini, and P. Samarati, "An XACML-based privacy-centered access control system," in *Proceedings of the First ACM Workshop on Information Security Governance*, ser. WISG '09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 49–58.

[56] M. J. May, C. A. Gunter, and I. Lee, "Privacy APIs: Access control techniques to analyze and verify legal privacy policies," in *19th IEEE Computer Security Foundations Workshop (CSFW'06)*, Jul. 2006, pp. 13 pp.–97.

[57] S. Pinisetty, T. Antignac, D. Sands, and G. Schneider, "Monitoring data minimisation," *arXiv preprint arXiv:1801.02484*, 2018.

[58] T. Pulls, R. Peeters, and K. Wouters, "Distributed privacy-preserving transparency logging," in *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, 2013, pp. 83–94.

[59] D. Gonçalves-Ferreira, M. Leite, C. Santos-Pereira, M. E. Correia, L. Antunes, and R. Cruz-Correia, "HS.Register – An audit-trail tool to respond to the General Data Protection Regulation (GDPR)," in *Volume 247: Building Continents of Knowledge in Oceans of Data: The Future of Co-Created eHealth*, ser. Studies in Health Technology and Informatics, 2018.

[60] R. Peeters and T. Pulls, "Insynd: Improved privacy-preserving transparency logging," in *Computer Security – ESORICS 2016*, I. Askoxylakis, S. Ioannidis, S. Katsikas, and C. Meadows, Eds. Cham: Springer International Publishing, 2016, pp. 121–139.

[61] R. Accorsi and C. Wonnemann, "Auditing workflow executions against dataflow policies," in *Business Information Systems*, W. Abramowicz and R. Tolksdorf, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 207–217.

[62] H. J. Pandit, A. Polleres, B. Bos, R. Brennan, B. Bruegger, F. J. Ekaputra, J. D. Fernández, R. G. Hamed, E. Kiesling, M. Lizar, E. Schlehahn, S. Steyskal, and R. Wenning, "Creating a vocabulary for data privacy," in *On the Move to Meaningful Internet Systems: OTM 2019 Conferences*. Cham: Springer International Publishing, 2019, pp. 714–730.

[63] B.-J. Koops and R. Leenes, "Privacy regulation cannot be hardcoded. A critical comment on the 'privacy by design'provision in data-protection law," *International Review of Law, Computers & Technology*, vol. 28, no. 2, pp. 159–171, 2014.