Enterprise Architecture Patterns for GDPR compliance

Clara Moleiro dos Santos Teixeira

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. André Ferreira Ferrão Couto e Vasconcelos

Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

Examination Committee

Chairperson: Prof. Daniel Jorge Viegas Gonçalves

Supervisor: Prof. André Ferreira Ferrão Couto e Vasconcelos

Member of the Committee: Prof. Alberto Manuel Rodrigues da Silva

January 2021

Abstract

With the growth of technology and the personalization and customization of the internet experiences, personal data has been stored and processed more and more. In some cases, the data subject has not agreed with the retrieval and the purpose of the processing. To solve this, the European Union (EU) parliament approved the General Data Protection Regulation (GDPR), a regulation that has the data subjects' interests in mind. Since some of the concepts and requirements are hard to comprehend, patterns can help system architects and engineers to deliver GDPR compliant information systems. It is important to emphasize that these privacy-related concerns should be addressed at a design level, not after the implementation. This methodology is mostly known as Privacy by Design. This work focuses on the requirements brought by the GDPR, especially on the requirements related to the data subject's rights, and in providing enterprise architecture patterns to achieve GDPR compliance by proposing a library of patterns. This library is organized in 11 use cases with the GDPR principles that they address; it has 22 patterns, two of which we adapted from others, and each one handling one or more use cases, modeled in ArchiMate, for a clearer understanding of the solutions. A template was created to describe the patterns, having the Context, Problem, and Solution addressed. These patterns focus more on the business level but also tackle matters at the applicational and technology level. The patterns were applied to a case study, and the impacts were assessed.

Keywords: GDPR, Compliance, Personal Data, Enterprise Architecture Patterns, Privacy by Design

Com o crescimento da tecnologia e a personalização de serviços na internet, os dados pessoais têm sido cada vez mais armazenados e tratados. Em alguns casos, o titular dos dados não concordou com a recolha e a finalidade do tratamento. Para resolver esta situação, o parlamento da União Europeia (EU) aprovou o Regulamento Geral de Proteção de Dados (RGPD), um regulamento que tem os interesses dos titulares dos dados em mente. Como alguns dos conceitos e requisitos são difíceis de compreender, padrões podem auxiliar arquitetos e engenheiros de sistema a fornecer sistemas de informação compatíveis com RGPD. É importante realçar que estas questões relacionadas com privacidade devem ser tratadas ao nível da criação do projeto, não após a sua implementação. Essa metodologia é conhecida como *Privacy by Design*. Esta dissertação foca-se nos requisitos trazidos pelo RGPD, especialmente nos relacionados com os direitos do titular dos dados, e no fornecimento de padrões de arquitetura empresarial para atingir o cumprimento do regulamento, propondo uma biblioteca de padrões. Esta biblioteca está organizada em 11 casos de uso com os princípios RGPD que abordam; possui 22 padrões, cada um endereçando um ou mais casos de uso, modelados em ArchiMate, isto para uma melhor compreensão das soluções. Um template foi criado para descrever os padrões, tendo o seu Contexto, Problema e Solução abordados. Estes padrões focam-se mais a nível de negócios, mas também abordam questões a nível aplicacional e tecnológico. Os padrões foram aplicados a um caso de estudo e os impactos avaliados.

Palavras-chave: RGPD, *Compliance*, Dados Pessoais, Padrões de Arquitetura Empresarial, *Privacy by Design*

Table of Contents

Abstract iii

Table of Contents v

List of figures vii

List of Tables ix

List of Acronyms xi

1. Introduction 1

1.1. Problem definition 1

1.2. Work objectives 2

1.3. Document structure 3

2. Background 4

2.1. GDPR 4

2.1.1. GDPR Principles 4

2.1.2. Rights of the Data Subject 5

2.2. Privacy by Design 6

2.2.1. Privacy by Design Principles 6

2.2.2. Privacy by Design Strategies 7

2.3. Patterns 8

3. Related work 10

3.1. Steps for GDPR Compliance 10

3.2. Existing Patterns 12

3.3. Existing Tools and Practices for GDPR Compliance 12

3.4. Analysis 16

4. Proposal 18

4.1. Solution Overview 18

4.2. Guidelines for GDPR Compliance 18

4.3. Entities and Use Cases 21

4.4. Template for Patterns 24

4.5. Retrieval of Patterns 25

4.6. Library 29

4.6.1. Register in system Use Case 29

4.6.2. Inform of Breach Use Case 46
4.6.3. Request for restriction on personal data's processing Use Case 52
4.6.4. Request of personal data Use Case 56
4.6.5. Request for portability Use Case 61
4.6.6. Request for erasure of data Use Case 63
4.6.7. Request for/and update of data Use Case 65
4.6.8. Change of data processing purpose Use Case 67
4.6.9. Consent of minors Use Case 68
4.6.10. Transfer data processing to a third-party Use Case 71
4.6.11. Notify the data subject Use Case 75
5. Demonstration 79
5.1. Description of Platform 79
5.2. Selecting Use Cases 79
5.3. Selecting the Patterns 80
5.4. Changes on the platform 83
6. Evaluation 88
6.1. Differences from other works 88
6.2. GDPR Principles addressed 89
6.3. Quality of the Library 90
6.4. Possible Benefits 90
6.5. Shortcomings 91
7. Conclusion 92
7.1. Contributions 92
7.2. Future work 93
8. References 94

List of figures

Figure 1 - Classifying data with Heat Maps in LeanIX [16] 13

Figure 2 - Connection between Privacy by Design Strategies and Requirements 14

Figure 3 - Proposed BPMN solution for registration process [20] 15

Figure 4 - Proposed model for compliance with article 12.2 of Directive 2016/680 [21] 16

Figure 5 - Proposed Solution Process 18

Figure 6 - Guidelines for new projects 20

Figure 7 - Entities and their relations 21

Figure 8 - Use Cases 22

Figure 9 - Diagram of Register in system Use Case 30

Figure 10 - Diagram of Lawful Consent 32

Figure 11 – Diagram of Informed Implicit Consent 34

Figure 12 - Diagram of Obtaining Explicit Consent 36

Figure 13 - Diagram of Minimal Information Asymmetry 38

Figure 14 - Diagram of Awareness Feed 41

Figure 15 - Diagram of Encryption with user-managed keys 42

Figure 16 - Diagram of Aggregation Gateway 44

Figure 17 - Diagram of User data confinement pattern 45

Figure 18 – Diagram of Personal Data Store 46

Figure 19 - Diagram of Inform of Breach Use Case 47

Figure 20 - Diagram of Data Breach Notification Pattern 48

Figure 21 - Diagram of Unusual Activities 51

Figure 22 - Diagram of Request for restriction on personal data’s processing use case 52

Figure 23 - Diagram of Negotiation of Privacy Policy 54

Figure 24 - Diagram of Reasonable Level of Control 56

Figure 25 - Diagram of Request of personal data use case 57

Figure 26 - Diagram of Personal Data Table 58

Figure 27 - Diagram of Privacy Dashboard 60

Figure 28 - Diagram of Request for portability use case 61

Figure 29 - Diagram of Privacy Dashboard (adapted)....62

Figure 30 - Diagram of Request for erasure of data use case 63

Figure 31 - Diagram of Technical Solution for Right of Erasure 65

Figure 32 - Diagram of Request for/and update of data use case 66

Figure 33 - Diagram of Technical Solution for Update of data....67

Figure 34 - Diagram of Change of data processing purpose use case 68

Figure 35 - Diagram of Consent of minors use case....69

Figure 36 - Diagram of Lawful Consent (adapted) 71

Figure 37 - Diagram of Transfer data processing to a third-party use case 72

Figure 38 - Diagram of Sticky Policies or Obligation Management....73

Figure 39 - Diagram of Trust Evaluation of Services Sides....75

Figure 40 - Diagram of Notify the data subject use case 76

Figure 41 - Diagram of Asynchronous notice 78

Figure 42 - Diagram for Sign up of the user....84

Figure 43 - Diagram for Suspicious login 84

Figure 44 - Diagram for Access history logs....85

Figure 45 - Diagram for Update the profile....86

Figure 46 - Diagram for Delete profile 86

Figure 47 - Access to other platforms....87

List of Tables

Table 1 - 7 Privacy by Design Principles [6] 6

Table 2 - Association of Hoepman’s Privacy by Design Strategies and GDPR Principles 8

Table 3 - [18] requirements specification 13

Table 4 - Pattern Sources and Good Practices 17

Table 5 - Selected Patterns 27

Table 6 - Lawful Consent 30

Table 7 - Informed Implicit Consent 33

Table 8 - Obtaining Explicit Consent 34

Table 9 - Minimal Information Asymmetry 36

Table 10 - Awareness Feed 38

Table 11 - Encryption with user-managed keys 41

Table 12 - Aggregation Gateway 43

Table 13 - User data confinement pattern 44

Table 14 - Personal Data Store 45

Table 15 - Data Breach Notification Pattern 47

Table 16 - Unusual Activities 49

Table 17 - Negotiation of Privacy Policy 53

Table 18 - Reasonable Level of Control 54

Table 19 - Personal Data Table 57

Table 20 - Privacy Dashboard 59

Table 21 - Privacy Dashboard (adapted) 61

Table 22 - Technical Solution for Right of Erasure 64

Table 23 - Technical Solution for Update of data 66

Table 24 - Lawful Consent (adapted) 69

Table 25 - Sticky Policies or Obligation Management 72

Table 26 - Trust Evaluation of Services Sides 73

Table 27 - Asynchronous notice 76

Table 28 - Relation between platform and GDPR library use cases 80

Table 29 - Use Cases and Patterns Selected..........83

List of Acronyms

EA	Enterprise Architecture
EU	European Union
GDPR	General Data Protection Regulation
IT	Information Technology
PIA	Privacy Impact Analysis

1. Introduction

The importance of securing clients' and employees' personal information has always been evident. However, the growth of technology and the need to ensure that data is safely stored required a common regulation [1]. So, the General Data Protection Regulation (GDPR), a regulation concerned with the personal data of the citizens, was created and applied. Although several countries already had some legislation regarding this issue, it was not the same for everyone; therefore, some countries had an easy adaption while others had to start from the ground. Companies and other organizations had to question: "How do we achieve GDPR compliance?", to answer this, the requirements brought by the GDPR and the steps needed for compliance were collected and analyzed. Nevertheless, only knowing what is new is not enough; what would be helpful is to know how to achieve this compliance. By reading the regulation, we have an idea about the changes but not a solution to address those changes, and here is where patterns appear.

According to Alexander [1], "each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice." and are used in different domains, like meta-programming, games, etc. Since patterns provide solutions for recurring problems and can be used multiple times, they are a perfect tool to solve the constraints brought by the GDPR. Unfortunately, in the regulation itself, these new concerns do not come with "how to's" for its implementation in projects and services; but the patterns can help. This work aims to identify relevant patterns that can provide solutions to the implementation problems and present them organized according to GDPR principles and requirements. Some work has already been in progress to help companies with this matter, as presented in chapter 3, but most are tools or work done for specific cases. Privacy by design is also a domain that is very connected to the matter and already has many patterns, but they are not organized to help with the specific case of GDPR.

In sum, GDPR is a very recent regulation that brought the attention of the companies that have to deal with it daily and the general population that uses services to which they provided personal data. With many voices talking about the same matter, it is hard to see what organizations need to do and how to do it.

Not all services are equal, so providing one very detailed solution is not a way to solve the problem. It is best to provide different ways to resolve a situation that can be adapted to the different realities of the services that deal with personal information.

1.1. Problem definition

Companies are now faced with new challenges and are not familiar with some of the terms and constraints the GDPR brings. This regulation gives extensive rules concerned with personal data and the data subject's rights related to it. However, its language may not be very familiar to the people who have to put it into practice. There may exist tools or frameworks that help companies understand better the different types of personal data. However, it is essential to know how to build services and systems that are GDPR compliant or what needs to be added and changed to older ones to achieve that compliance.

So, what exactly are these changes? In what cases is the GDPR applied, and how can we solve the problems related to personal data security?

This work is relevant since the GDPR is not a set of guidelines that organizations can choose to follow or not, but legislation concerned with the citizens' best interests. Not compliance with it can require companies to pay fines.

People are already aware of the regulation but may not know exactly its purpose nor how to implement these requirements in their services. This work is very relevant, not only for companies to understand better what needs to be done for GDPR compliance and solutions for how to do it but also to protect them from harmful practices and assure conformity with the legal regulation.

1.2. Work objectives

The goal is to create a GDPR organized library of patterns that help solve problems related to the GDPR requirements. The patterns are organized by use cases. These use cases are situations organizations may face related to GDPR constraints; they also assist in searching for the patterns relevant for the service they will be applied to. There are eleven use cases, five of which are related to the data subject's rights. The others address other situations, like registration, inform of breach, change of processing purposes, transfer processing to a third-party, child's data processing, and notification of the data subject.

One of the bases of patterns, as will be expressed later, is the template. For this work, the template is the name of the use case, a brief description of it (with a diagram), and the GDPR principles associated. Next is the name of the pattern, the context situation, the problem that arises from it, and the problem's solution. The source from where the pattern was adapted or retrieved is also present and a diagram in ArchiMate describing simply the processes or architecture present in the solution.

This library seeks to help companies find solutions that provide compliance to the GDPR by showing the overall constraints the regulation enforces and proposing solutions to them. A simple and more concise guideline was made to show some steps new projects have to follow for compliance with the GDPR. Those not familiar with the regulation or have doubts about what needs to change have their questions answered with this. The guideline does not provide compliance solutions, like the library, but it can be used as a guide and a checklist.

The work done focuses on providing patterns that can be used to their fullest or at least adapted to the services' circumstances. Not all use cases may be relevant to all services, but it is essential to try and provide solutions to general problems that arose with the implementation of the GDPR.

1.3. Document structure

This document is structured into seven chapters. In the next chapters, some background and related work are described, including GDPR principles, Privacy by Design, Patterns, as well as existing tools and practices for GDPR compliance. Chapter 4 shows the solution's proposal with an overview of it, followed by the definition of the solution's approach and the solution itself. Next, in chapter 5, a demonstration of how the library can be applied using a case study is presented. Chapter 6 presents a discussion of the proposal, with the benefits and downsides of the solution, and in the last chapter, we conclude and present the future work.

2. Background

In this chapter, some background will be presented in order to understand better the concepts approached in this work.

2.1. GDPR

The General Data Protection Regulation (GDPR) is a standardized and enforceable law across all EU Member States [2], allowing citizens to understand “how” and “what for” their data is being used. In simple terms, this regulation applies to any person, the data subject, in the EU whose data is being processed by an organization (e.g., legal person, public authority, institute, etc.) that operates within the EU, whether the processing is done in or outside of the European Union (Art. 3 [3]).

In 2015-2016, the European Parliament discussed the future regulation, and finally, in 2016, the GDPR and the EU Directive were published in the Official Journal of the EU. More proposals and corrections are made to the regulation in the years to come, and finally, in May of 2018, the General Data Protection Regulation is applied. It consists of 11 chapters and 99 articles [4].

The regulation has terms like Data Controller, which is “the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data” (Art4, paragraph 8) [3]. Moreover, it enforces Consent, “any freely given, specific, informed and unambiguous indication of the data subject’s wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her” (Art4, paragraph 11) [3].

2.1.1. GDPR Principles

The GDPR brings a set of principles that are related to the processing of personal data. EXIN [5] resumes them in 6 principles, which we can find in Article 5 [3].

The principles of lawfulness, fairness, and transparency (paragraph 1, sub a GDPR) are further explained in Articles 7, 8, 9 GDPR and describe that personal data processing should be done according to the law. It should be transparent (Art 12) [1] with the data subject in terms of data processing (which data is processed and for which purpose), breaches, and provide access to their data. The term fairness is related to compliance with the legislation and with the data subject rights.

Principle of purpose limitation (paragraph 1, sub b GDPR): this principle focuses that all collected personal data should be stored and processed for defined reasons and must only be used for those purposes. If the purposes change, another consent must be made.

Principle of data minimization (paragraph 1, sub c GDPR): only collect and process data that is relevant and strictly necessary; for example, if we only need the name, we do not have to store the age.

Principle of trueness and accuracy (paragraph 1, sub d GDPR): the data controller must ensure that the stored data comes from a legitimate source (that it is “true”) and must update or remove any inaccuracy.

Principle of storage limitation (paragraph 1, sub e GDPR): data that permits identification must be stored only for the amount of time needed and previously defined with the data subject. The data controller is responsible for tracking the data and removing it when it is no longer being processed for its original purpose.

Principle of integrity and confidentiality (paragraph 1, sub f GDPR): these principles are part of information security. Integrity is related to the data’s accuracy and consistency (the data subject maintains its integrity through their data). Confidentiality is about protecting the data from non-authorized access with the necessary measures.

Furthermore, in Art. 5(2) there is another, the principle of accountability, which states that the controller is responsible for compliance with paragraph 1 and must demonstrate it.

2.1.2. Rights of the Data Subject

In chapter 3 of the GDPR, the regulation describes the data subject’s rights that controllers need to ensure to comply with the regulation. Articles 13 and 14 express what needs to be informed to the data subject when the data is collected. Article 22 discusses automated individual decision-making, and Article 22 the restrictions. For this work, we will focus on Articles 15 to 21.

Right of access by the data subject (Art. 15) states that the data subject has the right to know if their data is being processed and access their data and additional information related to the processing.

Right to rectification (Art. 16) refers to having the data subject’s personal data accurate, allowing rectification of any inaccuracy.

Right to erasure (‘right to be forgotten’) (Art. 17), probably the most discussed, is the right to have their data removed from processing; this request can be applied depending on grounds in Art. 17 (1) and restrictions to it, Art. 17(3).

Right to restriction of processing (Art. 18) states that the data subject can restrict the processing of data in some grounds, Art. 18 (1), and that, although it can remain stored, it must only be processed for specific cases, Art.18(2).

Notification obligation regarding rectification or erasure of personal data or restriction of processing (Art. 19), the controller must communicate any rectifications, erasures, or restrictions to the data subject.

Right to data portability (Art. 20) refers to the right of receiving their data or having it sent directly to other controllers, in a “commonly used and machine-readable format”, Art. 20(1).

Right to object (Art. 21) is the data subject’s right to object to the processing of their data. It also discusses that the processing for marketing purposes must be explicitly informed to the data subject and must be separated from other purposes, Art. 21(2,3,4). An everyday example of this request is the use of cookies.

2.2. Privacy by Design

Privacy by Design is about considering privacy when designing systems and relates to GDPR Compliance because some of its principles are similar to the legislation's requirements. EXIN Privacy & Data Protection states that: "required level of data protection must already be taken into account at the design stage for the processing method" [5].

2.2.1. Privacy by Design Principles

[6] names the principles in the foundation of this approach, and here we can see the similarities between these principles and the GDPR principles.

Table 1 - 7 Privacy by Design Principles [6]

First – Proactive and Preventive
Second – Privacy as the Default
Third – Privacy embedded into Design
Fourth – Full functionality (Positive-Sum)
Fifth – End-to-end Security
Sixth – Visibility and Transparency

The first principle refers to the preventive and proactive measures since the focus is not waiting for breaches or risks to happen to fix them but preventing them from happening in the first place, showing a commitment to privacy. Privacy by Default is the second principle. It focuses on the data being "automatically" secured when it enters the system by following requirements (some of which we can find in the GDPR).

The third principle is about embedding privacy into the design and not for it to be an add-on. This can be accomplished using existing standards and frameworks as guidelines and providing privacy impact and risk assessments. The fourth principle refers to accommodating non-privacy goals and creating a win-win situation by satisfying both privacy and non-privacy concerns (often rejecting trade-offs). The fifth principle is about security throughout the information's lifecycle (from start to finish).

In the sixth principle, the focus is on transparency and visibility, ensuring that all business practices follow what was promised, and the services are visible and transparent to users and providers. The last principle is related to the user’s privacy, the individual’s privacy, and this should be of uppermost concern.

2.2.2. Privacy by Design Strategies

[7] divides the requirements into strategies, which are “architectural goals in privacy by design to achieve a certain level of privacy protection”. These goals are:

- Minimize: limit the data to only the essential for our system, reducing the breach impact.
- Hide: use of cryptography and restrict access to only authorized personnel, helping reduce the probability of a breach.
- Separate: distributing or isolating storage also helps in reducing the probability of a breach.
- Abstract: limit the detail of information, reducing the impact of a breach.
- Inform: inform the data subject of changes, requests, retention of the data, and notify them when a breach occurs.
- Control: the consent to, update, and retract data from the data subject, control over their personal data.
- Enforce: ensuring the commitment to the GDPR requirements, policies, and legislation by updating and chasing the wrong practices.
- Demonstrate: having evidence of the compliance with GDPR by having logs and audits to extract better the goals and effects of the actions performed on personal data.

Table 2 - Association of Hoepman's Privacy by Design Strategies and GDPR Principles

GDPR Principles

		Purpose Limitation	Data Minimizations	Trueness, Accuracy	Storage Limitation	Integrity and Confidentiality	Lawfulness	Fairness	Transparency	Accountability
Strategies	Minimize	x	x							
	Hide				x	x				
	Separate					x				
	Abstract				x					
	Inform			x					x	
	Control						x	x	x	
	Enforce									
	Demonstrate									x

Table 2, shows how the GDPR principles and Hoepman's Privacy by Design strategies can be associated.

When designing the systems, it is necessary to keep in mind these requirements and principles to achieve GDPR compliance. The principles and strategies are great checkmarks and guidelines for the work.

2.3. Patterns

As mentioned in the introduction, many use patterns to solve recurrent problems in an outlined way. Another definition, by Alexander, is: "The pattern is, in short, at the same time a thing, which happens in the world, and the rule which tells us how to create that thing and when we must create it. It is both a process and a thing: both a description of a thing which is alive, and a description of the process which will generate that thing." [8].

The book [9] defines: "A pattern describes a particular recurring design problem that arises in specific design contexts and presents a well-proven solution for the problem. The solution is specified by describing the roles of its constituent participants, their responsibilities and relationships, and the ways in which they collaborate". In conclusion, a pattern addresses a recurring design problem that arises in specific design situations and presents a solution.

In [10], the bases of what makes a pattern is defined as:

- the context (a situation giving rise to a problem)
- the problem (the recurring problem arising in that context)
- the solution (a proven resolution of the problem).

Patterns can have more features to define them, but these are the core of patterns. There are many examples of patterns in several domains, and since we are looking for patterns, examples of them will be shown in the next chapters.

3. Related work

In this chapter, a brief report of some of the works made around this subject as well as a short analysis of them.

3.1. Steps for GDPR Compliance

Since GDPR brings new terms and requirements, it is not easy for organizations to keep up and know what they need to do to achieve GDPR compliance, especially those that previously did not show interest in privacy matters. To help companies to understand what they need to do, researchers defined steps for GDPR compliance.

The blog [2], defines the following steps:

1. Identify (classify this data with respect to its privacy sensitivity)
2. Inform (Detail the purpose for which this data was collected, and ensure you possess or obtain the consent of the data subjects to use it in that way)
3. Analyze (Which applications, processes, people, and parties use this data, at which locations, for which purpose?)
4. Define controls (define controls and mitigating measures, using widely referenced standards such as the ISO/IEC 27001 as a basis for identifying useful controls)
5. Implement (Implement the controls and measures you have defined in your organization, processes, and systems, and test their security)

[11] states key steps to GDPR compliance as:

1. Establish an accountability and governance framework
2. Create a project team. Scope and plan the project
3. Conduct a GDPR gap analysis
4. Conduct a data inventory and data flow audit
5. Develop operational policies, procedures, and processes
6. Communication
7. Monitor and audit compliance

For Lankhorst [12], the steps for compliance are:

1. Teaming up with these officials and making them aware of the potential contribution of architecture is the first step
2. Creating a “privacy inventory” (Identify all data that counts as personal, classify this data with respect to its privacy-sensitivity, describe the purpose for which this data was collected, and ensure you have (or obtain) the consent of the data subjects to use it in that way and pay extra attention to special categories of personal data)
3. Analyze the use of personal data

4. Assess risks to sensitive data, in particular concerning the rights and freedoms of data subjects
5. Define controls and mitigating measures.
6. Prioritize risks, allocate budgets and plan the requisite changes and improvements
7. Implement the controls and measures
8. Demonstrate compliance to the regulatory authorities

In [13], the steps present are as follows:

Step 1 - Awareness and accountability
Step 2 - Scope the project and create a data overview
Step 3 - Conduct a GDPR compliance assessment
Step 4 - Propose areas of improvements based on assessment
Step 5 – Audit, revise and repeat

After reading the GDPR document and the papers [2] [11] [12] [13], we can summarize the steps for compliance with the GDPR into these:

1. Define the Data Protection Officer (DPO)
2. Perform a GDPR gap analysis
3. Identify the data
   a. Data used vs. data needed
   b. Classify data (personal data, special categories)
   c. Analyze usage (access restrictions, consent)
4. Risk analysis (PIA)
5. Define controls and mitigation measures
6. Develop and implement the controls
7. Monitor the compliance (via audits, for example)

As we can see, [11] focuses on GDPR compliance for new systems, while the others focus on existing systems [2] [12] [13], so we tried to have the best of both worlds, but each organization should look at the steps and adapt them to their reality. For example, in terms of data, suppose we are looking at an existing system. In that case, it is essential to identify the data that was collected and is still stored versus the data that is strictly necessary to store. However, if we are looking for a new system, we only have to focus on what is necessary to store.

Throughout this entire process, it is crucial to point out that it is imperative to instruct the employees, partners, and clients of the upcoming changes since many risks and security breaches come from humans.

This research's focus is on the fifth step of the summary. However, we also have to consider all the previous and next steps to have a complete view of architectural needs.

3.2. Existing Patterns

One of the many domains that patterns can be applied is Privacy by Design. To better comprehend this subject, previous works and studies were analyzed. In [14], three sample patterns are provided from *privacypatterns.org*. One of them has a strong correlation to GDPR compliance, Location granularity. It is described as: Collecting more information than needed can harm the user's privacy and increase the risk for the service (in the case of a security breach, for example), but the location data may still need to be collected to provide the service[1]. The other two, Asynchronous notice and Privacy Dashboard, may not appear to be relevant to GDPR compliance at a first read, but they do, and in fact, they are used in the library.

This project's (*privacypatterns.org*) goal is "for this to be a living document constructed by the community of engineers, designers, lawyers and regulators involved in this topic"[2]. So, since the publication of [14], more patterns were added. In this website, the patterns are divided by Privacy by Design strategies and are generally defined by Summary, Context, Problem, Solution, and Consequences. This library is very relevant to the proposed solution since twenty of the selected patterns come from this source.

In 2017, a literature study was conducted on privacy patterns; in this research, the authors found a lack of studies focused on pattern catalogs since some were quite specialized [15]. In the study, the authors state that "the published research results show a clear focus on the privacy design strategies of hide and separate" [15]. No patterns were provided in [15] since the goal was to characterize and classify the different researches on this topic.

3.3. Existing Tools and Practices for GDPR Compliance

With the emergence of the regulation, many companies started to provide frameworks, like LeanIX [16], an Enterprise Architecture (EA) and Governance Tool that helps the companies in categorizing data objects in terms of privacy sensitivity, identifying responsibilities, classify the data in heatmaps, and many other concerns.

[1] https://privacypatterns.org/patterns/Location-granularity

[2] https://privacypatterns.org/about/

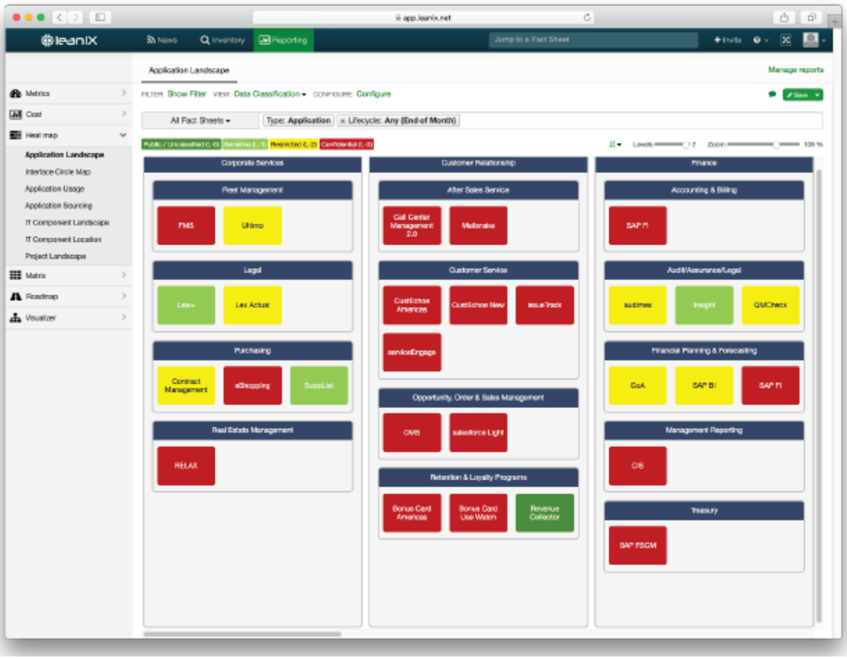


Figure 1 - Classifying data with Heat Maps in LeanIX [16]

Figure 1, shows a window of the framework that displays the data classified by heatmaps.

The PDP4E [17] is a project that aims to "widespread the creation of products, systems and services that better protect the privacy and personal data of EU citizen". PDP4E presents some papers and have participated in conferences about risk management[3], privacy-aware design[4], and other topics. The PDP4E project focuses more on tools and GDPR/privacy awareness, so no specific solution is provided.

A practical and design-oriented approach in order to solve GDPR's requirements is provided in [18]. The article divides the requirements and principles mentioned above into nine requirements that a system should take into account in its architecture: system security and privacy, data minimization, consent control, data traceability, user access, data rectification, data erasure, data restrictions, and data's physical location.

Table 3 - [18] requirements specification

R1: System security and privacy	1.1 appropriate data protection measures 1.2 confidentiality 1.3 integrity 1.4 availability 1.5 resilience 1.6 timely restoration 1.7 process for testing privacy
R2: Data minimization	
R3: Consent Control	3.1 collecting consent 3.2 consent data model 3.3 consent of guardians

[3] https://www.pdp4e-project.eu/risk-management/

[4] https://www.pdp4e-project.eu/privacy-aware-design/

R4: Data traceability	4.1 log of processing events 4.2 log of GDPR requests 4.3 log of moving data outside EU 4.4 log of third-party disclosures
R5: User access	5.1 viewing registered data 5.2 user interface for access requests 5.3 access to machine readable data (portability)
R6: Data rectification	6.1 updating personal data 6.2 user interface for update requests
R7: Data erasure	7.1 deleting personal data 7.2 user interface for update requests
R8: Data restrictions	8.1 restricted data 8.2 user interface for objections
R9: Physical location of data	

More detail for each requirement is in the table above.

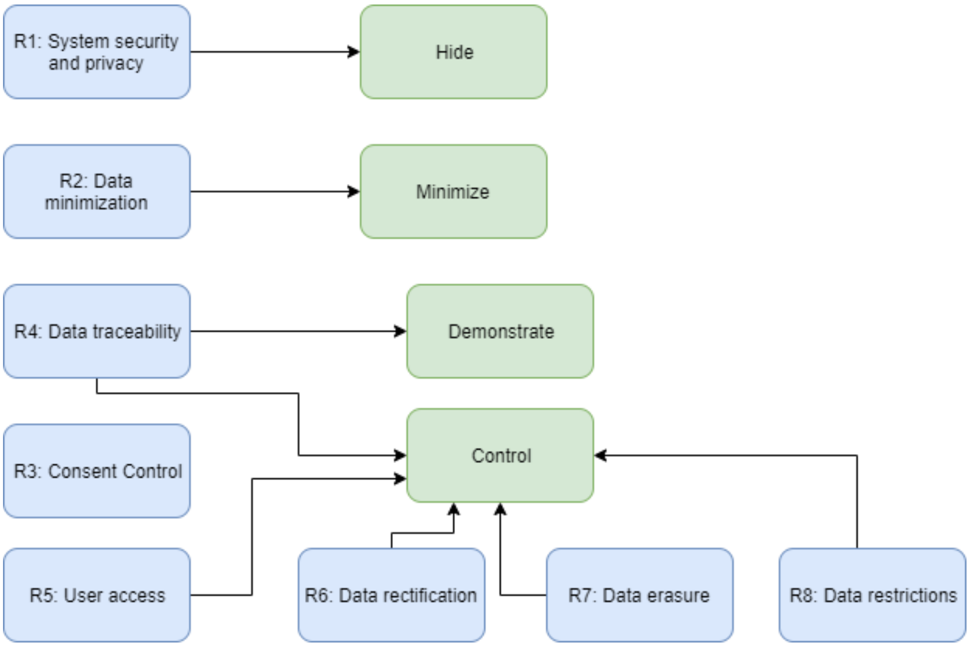


Figure 2 - Connection between Privacy by Design Strategies and Requirements

We can see the connections between some of the strategies presented in privacy by design and the requirements in Figure 2.

This approach mentions the logs, databases, and some application components needed for the architectural components from the GDPR requirements identified above, like web application server and an interface, services for GDPR request, business logic, personal data event log, and others. The work focuses more on detailed information architecture.

[19] provides patterns (technical solutions) for some of the GDPR principles or requirements (as defined in the paper) and data subject's rights. For example, in terms of personal data storage, the regulation states that personal data must be stored in a form, which permits the identification of the data subjects only as long as it is necessary for the original purposes of processing (Article. 5(1)e). The technical solution approach the authors present suggests having a data model that includes a data lifecycle. The lifecycle is based on time and attributes that declare a processing purpose. If the data is encrypted, an irreversible deletion of the key is enough to make the data non-identifiable. The authors also propose specific architectures and interfaces in some patterns, such as for restriction of processing (where a data subject has the right to restrict what personal data can be processed).

This paper is very relevant to this research but lacks modeling, and it is incomplete, as they mention in the paper; so, these patterns will be kept in mind for the solution but do not satisfy what is needed.

A BPMN proposal for a better understanding of the requirements brought in with the GDPR is created in [20]. The authors' approach involves defining a use case (a simple BPMN), gather authorization requirements, business requirements, and security best practices. Then an identification of the business process affected by the GDPR requirements is performed, and the statements are transformed into machine-interpretable language. The final steps are the test of the architecture, its deployment, the policies, and, at last, an access review.

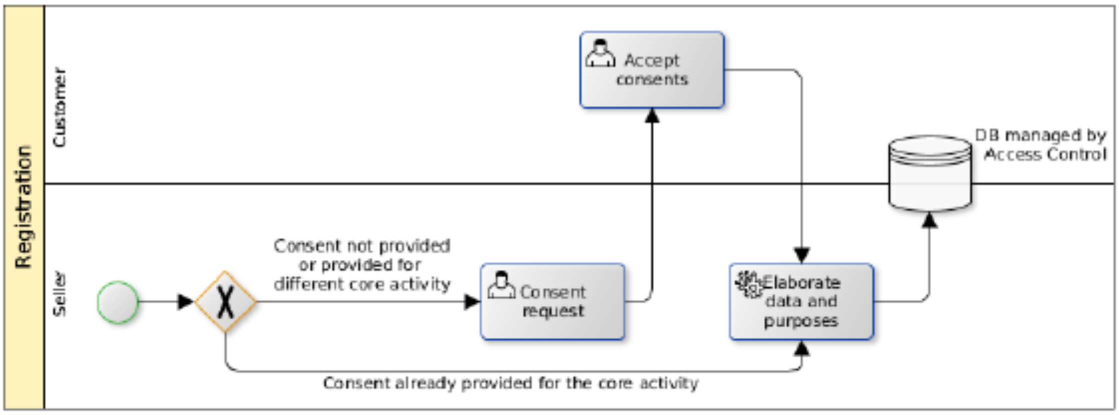

Registration
Customer
Seller
Accept consents
DB managed by Access Control
Consent not provided or provided for different core activity
Consent request
Elaborate data and purposes
Consent already provided for the core activity

Figure 3 - Proposed BPMN solution for registration process [20]

The authors provide a simple example where they select a business process, and then for each activity in that process, a list of GDPR requirements is made. A new model is created by adding the needed activities and sub-processes (Figure 3).

The step approach here is a promising start-up for the solution since it is easier to look at the requirements one at a time and then join them, but the BPMN approach is not what we are seeking.

[21] proposes an architectural meta-model for the EU Directive (Directive 2016/680). It is a directive concerned with the protection of people regarding the processing of data, created in April of 2016. This model was made taking into account that directive. Although both regulations concern processing data, they are not the same, so it has some differences in requirements and constraints.

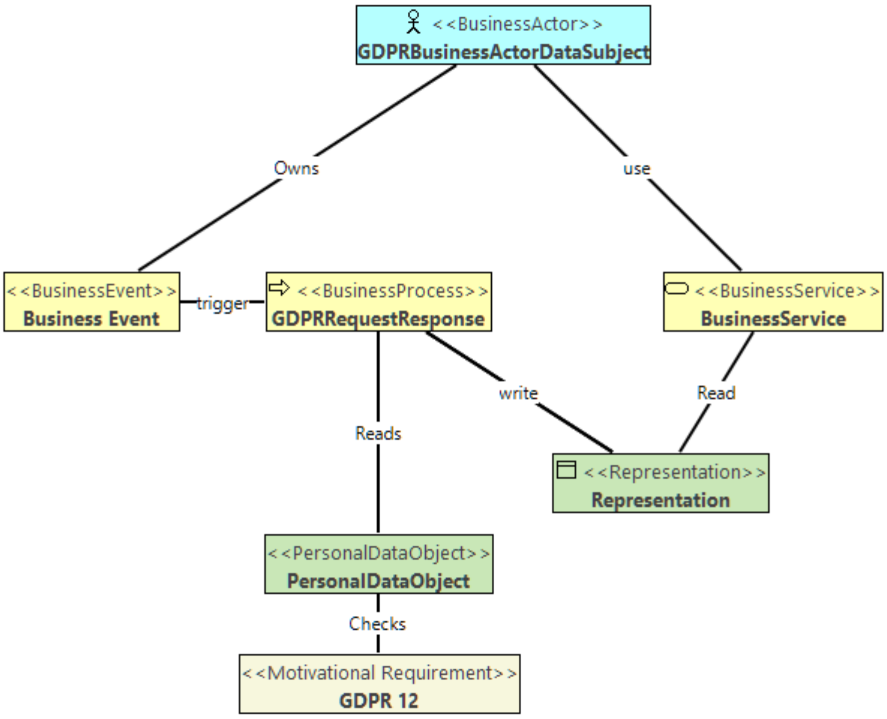


Figure 4 - Proposed model for compliance with article 12.2 of Directive 2016/680 [21]

For each requirement, the author created a model in ArchiMate with the business events, processes, data objects, actors, etc., involved (Figure 4) and a pseudo-code to better understand the articles and requirements. Finally, as an example, a few requirements models were combined, and the authors created an overview of a Swedish company's Business Intelligence solution approach to the regulation. This model could be used more to check if GDPR compliance is guaranteed than being a guide for creating GDPR compliant systems.

3.4. Analysis

To summarize, leanIx [16] is a framework that helps companies to achieve GDPR compliance by their heatmaps and data flows. However, it does not show what is needed in terms of enterprise architecture. Instead, it does that job for the user. The other researches provide more guidance on building GDPR compliance systems; [18] gives a design-oriented approach, providing what requirements need to be in mind and some architectural solutions (like logs), but it does not provide patterns.

On the other hand, [19] provides patterns and technical solutions for some GDPR requirements, like storage limitation, but does not cover all of them. It is more focused on each principle and rights of the data subject separately and is not complete. BPMN modeling is done in [20] for the use cases used in an organization, ensuring that they are GDPR compliant, but they present are no patterns. Paper [21] presents models in ArchiMate, but some mismatch occurs because it follows the EU Directive 2016/680 and not with the General Data Protection Regulation; also, a map of the architecture is modeled and not a pattern.

Some researches lack modeling, while others lack a more pattern-oriented approach, and others are incomplete; nevertheless, all the learnings acquired when assessing these documents were considered when proposing and creating the final solution. The goal of this work is to create a library that guides companies and provides solutions in order for them to achieve GDPR compliance, so patterns from *privacypatterns.org* (the website referenced in [14]) and from [19] were considered to be part of the library.

The table below describes the sources for the patterns and good practices to consider.

Table 4 - Pattern Sources and Good Practices

	What to take	Benefits	Source
Patterns' sources	Privacy by Design patterns	-	[14]
	GDPR Patterns	-	[19]
Good Practices	Focus on the privacy by design strategies that are related to the GDPR requirements	Instead of looking through patterns in all strategies, this helps in restricting the search.	(Figure 2)
	The 9 requirements that a system should consider in its architecture.	When defining the use cases is easier to model the architecture needs.	[18]
	The step approach used, but with adaptations.	The benefit is looking for patterns for a specific scenario instead of looking without a use case.	[20]
	Use proposed model to check for architecture compliance.	Throughout the solution is important to make sure that GDPR compliance is being assured.	[21]

4. Proposal

This research aims to create a library of information and applicational architecture patterns for GDPR compliance. As presented in section 2.3, a pattern addresses a recurring problem, which in this case, it is GDPR compliance, and presents a solution. Therefore, this research proposes enterprise architecture patterns, mainly Privacy by Design patterns, that are expected to help organizations build information systems smoothly and with fully aware of the security practices and constraints needed to be compliant with the GDPR.

4.1. Solution Overview

As presented previously, there are already patterns in the Privacy by Design domain. However, a collection of patterns organized in terms of the General Data Protection Regulation principles and the data subject's rights intertwined does not exist.

Another particularity of the proposed solution is the definition of use cases. This approach is expected to make it easier to search and find which patterns make sense for each case (since not all the patterns need to be applied to all projects). Also, we based most of the use cases on the data subject's rights, providing a connection between these rights and the principles relating to the processing of personal data.

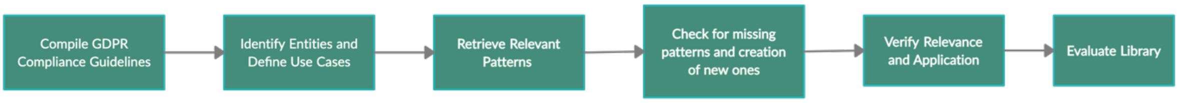

Compile GDPR Compliance Guidelines
Identify Entities and Define Use Cases
Retrieve Relevant Patterns
Check for missing patterns and creation of new ones
Verify Relevance and Application
Evaluate Library

Figure 5 - Proposed Solution Process

Before starting, we will **compile a few guidelines for GDPR Compliance** to go along with the library that can be used as a checklist or introduce the regulation. The proposed solution starts by **identifying the entities** (stakeholders and objects), then proceeds to **define the use cases** by analyzing the business processes needed, select the GDPR principles associated and the entities present, and later, if possible, model them.

For each use case and its principles, **relevant patterns are retrieved** from sources *privacypatterns.org* and [19] and adapted to our template. We will then **check** if all use cases have at least one pattern associated, and if not, we create or adapt a solution for it.

After creating a GDPR pattern library, we **verify** if the patterns are relevant and applicable to a Case Study. After the application of the patterns, the last step is their **evaluation**.

4.2. Guidelines for GDPR Compliance

Before we start describing the proposed library of patterns, we were requested to present guidelines related to the GDPR. Since the work focuses on assisting organizations to comply with the GDPR, and one of the problems is the lack of understanding of the regulation, we created this list. A brief guideline of actions related to the data that will be processed and considered in creating a new project is presented below.

- Define which data will be kept and processed. Be especially aware if the data is:
    - Personal data (Art. 4 (1)): information that identifies or may identify the user ("data subject") directly or indirectly. Examples: name, an identification number, data related to localization, etc.
    - Sensitive data or of special categories (Art. 9): "personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation shall be prohibited." (See restrictions in Art. 9 (2)))
- Define the purpose to which the data will be stored and processed:
    - If the purpose is to comply with legal obligations or of public interest (see Art.6 for better clarification and other reasons for lawful processing), consent is not required.
    - For other purposes, consent is needed.
    - If there is no purpose para the storage of the data, then its collection is not required.
- Define the time for which the data will stored (be careful and aware of legal requirements)
    - Formulation and request of consent (Art. 7):
        - The purposes have to be explicit,
        - Use clear and direct language,
        - It should be given freely, and give the possibility to reject,
        - It needs to show the term for which the data will be stored.
    - *As mentioned above, if the data is collected for legal, contractual obligations, health concerns, consent is not required (Art.6, (1) b,c,d,e,f)
- Storage (besides data itself):
    - Consent,
    - Time for which the data will be stored,
    - Purpose(s),
    - Accesses (who can access the data),
    - Security measures.
- Definition of how the data will be stored:
    - If sensitive data is stored, encryption is required,
    - Define and indicate how the data will be stored (like encryption, separation of data, etc.),
    - Indicate if any pseudonymization will be used.
- Verification and rectification of the collected data:
    - The data subject must be given the possibility to rectify incorrect data that concerns them (Art. 16),
    - If the data was not collected from the data subject, their lawfulness must be verified, and special attention must be given to Art. 14.
- Logs:
    - A register (logs) of the data must be kept:
        - Accesses,
        - Rectifications,
        - Restrictions to processing.
- The service must meet the data subject's rights:
    - It must be given to the data subject the possibility to:
        - Access the stored data related to them (Art. 15),
        - Rectify their incorrect personal data (Art. 16),

  - “Be forgotten”, erasure the data (Art. 17) *be aware of any legal obligation or requests without valid reason that can invalidate the erasure,
  - Restrict the processing of data (Art. 18) * be aware of any legal obligation or requests without valid reason that can invalidate the restriction,
  - Be notified of erasure, restriction, rectification, etc. of their data (Art. 19),
  - Request for the portability of their data (Art. 20) in a structured, commonly used, and machine-readable format (ex: Excel).
- **To point out that there are more restrictions and concerns related to the processing of personal data, like the case of minors, subcontracting, and transfer to countries outside of the EU, that are not addressed in this guideline.

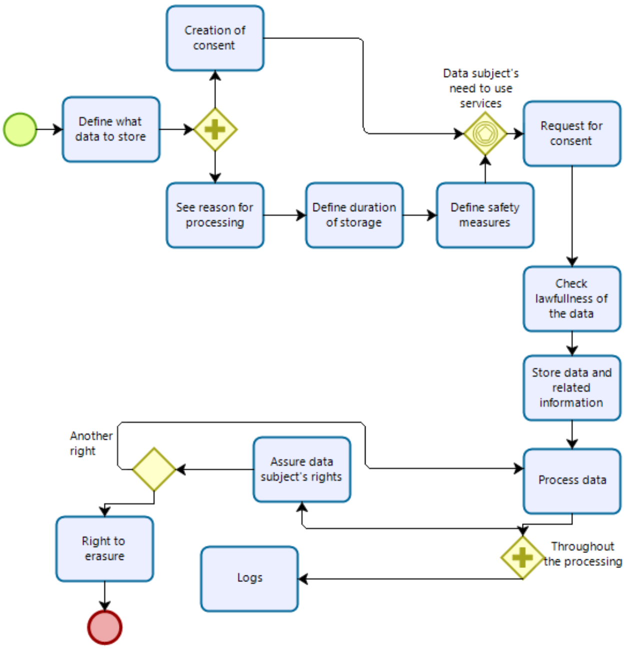

Creation of consent
Define what data to store
Data subject's need to use services
Request for consent
See reason for processing
Define duration of storage
Define safety measures
Check lawfullness of the data
Store data and related information
Another right
Assure data subject's rights
Process data
Right to erasure
Logs
Throughout the processing

Figure 6 - Guidelines for new projects

In Figure 6, we have a clearer view of the processes required to follow this guideline.

First, it is essential to define what data will be stored and processed, and it is vital to see if it is personal data (Art 4(1) [3]) or is a special category of personal data (Art. 9 [3]) since it brings more privacy concerns. Then, the purpose of why the data will be stored and processed should be stated. Not only to see if a need for explicit consent is required but also to understand if the processing of it is necessary to the service. Consent is not the only way to make the processing lawful, so before requesting it, one must check the other cases in Art. 6(1)b-f [3]. With this, the data storage duration (being aware of existing legal requirements) and security measures for it (encryption, pseudonymization, for example) must be defined.

Before the data subject starts using the services, the creation and request consent for each purpose should occur, again if applicable (Art. 7 [3]). In some cases, since some data may not be provided directly by the data subject, it is crucial to check the data’s lawfulness. Also, if new purposes appear and are different from those consented to, new consents are required. If new consents are not requested, and the processing continues, the data’s lawfulness is also in jeopardy.

After the data is collected, it must be stored with the previously defined security measures, the consent for each purpose, the storage period, the goal/reason for processing, and who has access.

Throughout the personal data processing, a record of logs of accesses, rectifications of data, and processing restrictions must be kept. The data subject's rights must be ensured, like the right of access, rectification restriction of processing, portability, and the others expressed in Chapter 3 [3]. When the data subject requests the erasure or the restriction of processing after a careful examination and the acceptance of the request, the processing of the data must cease.

With these guidelines, we can see the requirements that need to be followed to comply with the regulation. However, it does not show solutions or precisely what needs to be performed to ensure the data subjects' rights and the constraints that come from them. Here is where the library of patterns plays a role.

4.3. Entities and Use Cases

To better organize the library, use cases were defined, and the entities present were selected. In this chapter, we see why these entities and the use cases were selected and a brief description of what the use cases address. For an organized and complete library, it is important to provide an easy way of reading it.

The entities identified are the data subject (who can be a child), the data controller, the data processor, the third-party, and the data subject's holder of parental responsibility, which for the rest of the dissertation will be expressed as guardian (or guardian of the minor).

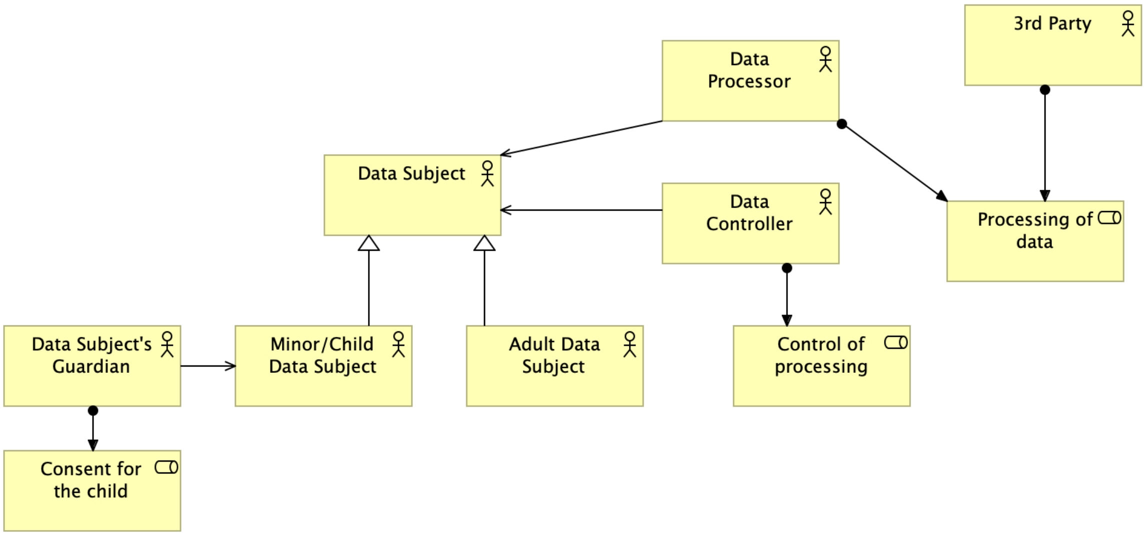


Figure 7 - Entities and their relations

Since most of the use cases are related to the data subject's rights, we can presume that the data subject will be present in almost all the use cases. This happens because of requests performed by the data subject itself, as the request for erasure, or by the controller, like a request for consent.

The data subject is the holder of personal data that can be identified by reference to that personal data. However, in this library, the data subject will only be the client/user of the organizations' services. In case the data subject is a child under the age of 16, we will call it minor.

It is also relevant to point out the difference between controller and processor since they are related. As explained in section 2.1, a controller is a person with legal authority that determines the purposes and means of personal data processing. A processor is a person who processes ("any operation or set of operations which is performed on personal data or on sets of personal data") the personal data on behalf of the controller. All these definitions are explained in depth in Article 4 of the regulation [3].

A third-party is an entity that is not a data subject, a controller, or a processor authorized to process personal data. It can even be from another country or even from outside of the European Union (EU).

All the entities related to a "minor" are for a child that, according to the regulation, is a data subject below the age of 16 years old, but the Member States can change it (although the age cannot go below 13 years) Art. 8 [3]. In Figure 7, we can see the relations between the entities and some of their roles.

The use cases selected are register in system, inform of breach, request for restriction on personal data's processing, request of personal data, request for portability, request for the erasure of data, request for/and update of data, consent of minors, update previous consent, change of data processing purpose, transfer data processing to a third-party (in or outside of EU) and notify the data subject.

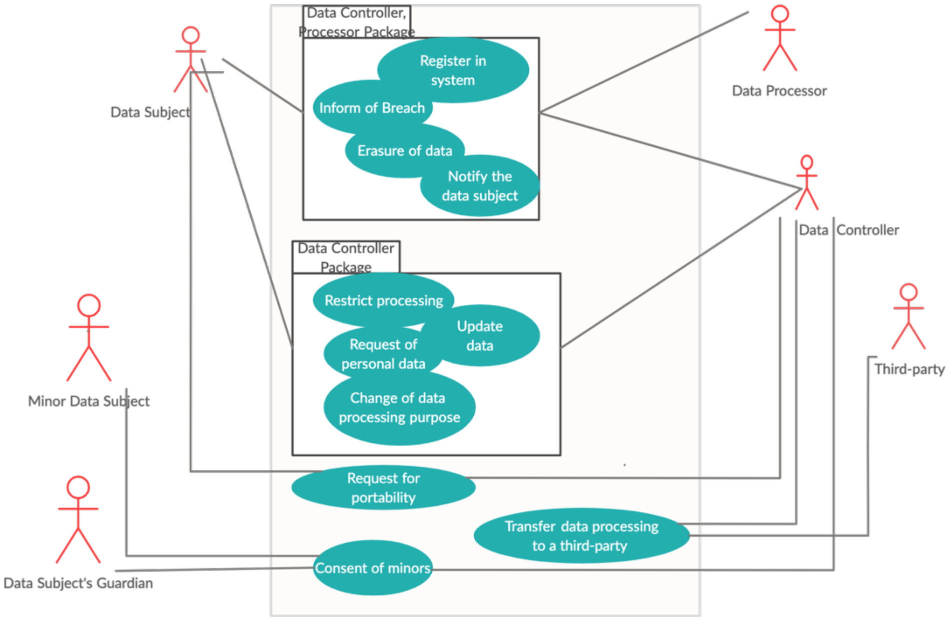


Figure 8 - Use Cases

Figure 8 is a UML Use Case diagram with all the use cases. The use cases with the same entities are grouped in packages for a clearer view.

Before describing the use cases, an explanation of their use is necessary. The use cases work as a way to organize the library per se and organize it in terms of the GDPR. Currently, we do not have a collection of patterns focused on solving the regulation problems that address both the principles related to the processing and the data subject's rights intertwined. The use cases are mostly situations that come from the data subject's rights, and in each use case, we also have the general principles that govern data processing.

The first use case is probably the most common and the one the general public is most aware of. When registering to a platform, the service can only process the data if the user, the data subject, consents to it or is part of the other lawful processing cases. As we will later present in section 4.6, this use case has many principles associated with it. First, the data controller has to define what the data will be used for; then, only the strictly necessary data is requested from the data subject with the related consent, if applicable. Finally, the personal data has to be stored and kept secure (with encryption, for example). If consent is required, it must be stored and be in a clear and straightforward language, assessing all the purposes for which each data will be used.

Informing a client of a breach is a use case that is probably already addressed by many organizations. However, the GDPR brings some requirements that probably were not contemplated before, like the controller notifying the supervisory authority of the breach within 72 hours and, if necessary, inform the data subject; it also has to follow the guidelines present in Article 33(3) [3].

The next use cases are particular to the regulation, mostly with the rights of the data subject, and are related to requests. The first is the request of restriction of the processing of the data subject's personal data; the second is the request of the personal data itself; the third is the request of portability; the fourth is the request of erasure; and, lastly, the request for/and update of personal data.

The restriction of data processing has some rules. For example, if the processing of the data is of public interest or necessary for legal purposes, the request for restriction is denied. However, if the data subject has reasons to believe that the processing is unlawful or is no longer needed, and they do not require its erasure, the restriction is accepted.

The other request is when the data subject asks for the data being processed and other information stated in Art. 15(1). We can comply with the request by accessing the logs referred to in section 3.3. A copy of the data should then be provided in a standard form. This use case also concerns the principle of transparency.

The request for portability is simple to explain, but not many people are familiarized with it. The data subject has the right to request the data from the data controller in “a commonly used and machine-readable way” (Art. 20 [3]). They can also transfer it to another controller or request the data to be directly transmitted, if possible.

On the other hand, the request for erasure, also known as “the right to be forgotten”, is probably more commonly known to the public than the previous right. The data subject has the right to request to the controller the erasure of their data if the processing is longer necessary or if it was made without their consent, for example. Of course, the request may not be accepted if the data is necessary for legal claims, the public interest, or other legal requirements. For example, in terms of billing, the client cannot request a store to immediately delete their shopping history since it may be necessary for the finance department.

Finally, in terms of use cases related to the data subject's rights, we have the request for or the ability to update their personal data. The data subject must be able to access their data quickly and update it; this may happen when an email or the residence changes. The rectification may also be to correct or complete false/incomplete information. All these cases need to have these functionalities visible and easily accessible to the users.

Another use case contemplates when new policies or purposes appear in the processing of the data. This requests for a new evaluation of what data will be necessary and the formulation of new consents for the new purposes. When the data is already being used for another, consented reason, the data subject must consent to the new purposes, or else the processing becomes unlawful.

As mentioned in the entities, a minor, in the regulation, a child younger than 16 years old, cannot consent for the processing; the same has to be given by a guardian, the holder of parental responsibility over the child. Since the age to give consent may differ between countries, it is essential to confirm it with each country's legislation.

Many companies subcontract others to process the data they collected, in or outside of the country or EU; when this occurs, it must be informed to the data subject (when requesting consent, for example). The processing can only occur if the third-party is considered trustworthy and if the controller or processor provided the appropriate safeguards. All principles must be associated with this use case since the third-party and the controller must follow the full extent of the regulation.

Lastly, we have a use case present in others, but since it is very relevant and can be used for other cases, it is on its own. This is the notification of the data subject. Whether it is a security breach, change of purpose, update, erasure, etc., the data subject has the right to be notified of all that concerns their personal information. With this transparency, not only do companies comply with the GDPR, but it can also offer the service provided as reliable and trustworthy.

The use cases were selected considering the broadest concerns and requirements of the regulation; more specific or industry-specific use cases are not addressed in this research.

4.4. Template for Patterns

One of the characteristics of patterns is their template, an explanation of what problem it addresses, and the proposed solution.

In this research, the template created has the base elements stated in [10] and elements present on *https://privacypatterns.org* patterns. The pattern template proposed considers what and how the pattern addresses the problem and the use cases for the search in the library.

The template for the patterns has the following fields:

Associated Use Case: The use case in which the pattern is applicable and a brief description of it (with a diagram of some of the general processes necessary).

Associated GDPR principles: The GDPR principles that the patterns aim to solve.

Name: Name of the pattern.

Context: The situation where the pattern may be applied.

Problem: The problem the pattern addresses.

Solution: The solution principle underlining the pattern.

Source: If the pattern exists in the accessible libraries, the source is included.

A simple diagram of the pattern is also present in the library to give a general idea of the pattern's solutions and requirements.

In the template, we see that, for each use case, GDPR principles are associated; this helps to see the main concerns of the problem, but it also helped search for the patterns. As was shown in Table 2, the principles are related to different Hoepman's Privacy by Design strategies; since these patterns are in the Privacy by Design domain, the strategies are some of the categories through which the search is done. However, not all the patterns for Privacy by Design are related to the GDPR, since the concern for privacy in the early stages of a "project" is prior to the creation of the regulation, and there are more problems related to privacy than the ones the GDPR brings. Some of the problems that patterns address are specific cases, like mobile applications, so a thorough and careful search was performed to find patterns that focused on more broad scenarios.

In each use case, the diagrams provide an overview of the constraints and requirements that come with the GDPR and show some of the "sub-processes" need to be taken into account. A diagram of the pattern is relevant; although not all solutions have architectural bases, a simple image type of view helps to see what is needed to be implemented clearly. A brief explanation of what it is and what articles in the GDPR are related to the addressed problems is also contemplated. This way, any person can analyze the regulation if they have a specific concern that is not covered in the library already with an idea of what to search for, making the search quicker.

A use case does not address only one principle. Consequently, it does not have only one problem associated with it, so the pattern's context is essential to see each situation (or process in the use case) we may apply the pattern. The problem and the solution are essentials for this type of library.

4.5. Retrieval of Patterns

When it came to retrieving patterns from the sources, not all use cases had the same ease. In this chapter, we present the list of the patterns, but first, some of the difficulties that came with the retrieval of patterns will be assessed.

One of the terms/concerns in the GDPR is Consent, and probably because it was the most visible change that companies and organizations had to comply to and it was already a known concept from its predecessor (i.e., Directive 95/46), a variety of patterns were found in this category. Not all could be used in the library because they are focused on particular cases, like the one presented above: Location granularity, which only focuses on the data subject's location-related data. Notification is also vastly addressed, but for many specific cases like pop-up notifications or icons for mobile applications that cannot be used in other platforms.

In contrast, patterns related to the data subject's rights were trickier to find since it is not a visible change, and many users may not be fully aware of its existence. For example, we found patterns that some companies have already used for the data subject requesting their data. They updated their security measures and provided the possibility to see the information online or download their activity (logs) or personal data. For the case of portability, it was harder to find patterns that addressed this matter, specifically when controllers provide the data directly to other controllers by the request of the data subject. This challenge was solved using the patterns of other use cases and adding the process of requesting to whom and how (email, for example) to send the information.

The erasure request also does not have many patterns. Although it may seem simple to erase data from a database, the GDPR also requests additional information to be stored and saved, like consents for all purposes, logs, and other information, not just the data itself. It is also needed to check if nothing has to be kept to fulfill legal constraints. The data subject has to be notified of the erasure or the resume of the storage. The concerns related to processing minors' personal data were also not easy to find; not only the child's age must be known, but the guardian also has to be aware of the processing and give consent. In this case, the controller has to decide if minors will use the service or not because if they will, age has to be a requested data. If not, it may not be relevant to process the data subject's age.

Two sources were analyzed to search for patterns that could solve the problems. Many patterns were retrieved from *privacypatterns.org*, but not from [19]; since they did not have as many patterns. The solutions were not in the needed structure, so they were adapted to fit the library template by adding the context and the problem. The solutions were also changed to fit the language and style of the other patterns.

For all use cases, the search on the website started by examining the requirements addressed and looking through the related Privacy by Design categories, the strategies. Then the patterns were selected after a brief reading of the patterns' context. Only the ones that were relevant for the use case were selected. Later the full extent of the patterns was read, and the most suited and embracing were chosen for the library. When the retrieval was more challenging, keywords like consent or deletion were also searched. Patterns referenced by others that were interesting for the work were also analyzed.

For use cases that did not have patterns in the sources that helped solve their problems, we noticed that we could adapt patterns from other use cases, so we did. Also, if a pattern could be used in more than one use case because it covers more than one GDPR principle, that pattern was added to the library for both use cases.

Below is the list of all the patterns and their use cases, accompanied by a brief explanation of what they address:

Table 5 - Selected Patterns

Use Case	Patterns	Brief Explanation
Register in system	Minimal Information Asymmetry [22]	The first two patterns are related to purpose limitation and data minimization, the next four are about integrity and confidentiality, and the last three are about consent.
	Awareness Feed [22]	
	Encryption with user-managed keys [22]	
	Aggregation Gateway [22]	
	User data confinement pattern [22]	
	Personal Data Store [22]	
	Lawful Consent [22]	
	Informed Implicit Consent [22]	
	Obtaining Explicit Consent [22]	
Inform of Breach	Data Breach Notification Pattern [22]	The first pattern focuses on quickly detecting and reacting to data breaches, and the second one is more related to authentication.
	Unusual Activities [22]	
Request for restriction on personal data's processing	Negotiation of Privacy Policy [22]	These patterns are about a data subject negotiating and being able to push and pull data for processing.
	Reasonable Level of Control [22]	

Request of personal data	Personal Data Table [22] Privacy Dashboard [22]	These patterns give the data subject the ability to see the data and logs and transfer the data to their computer.
Request for portability	Personal Data Table (adapted) [22]	These patterns are primarily for portability for the data subject, so the possibility of sending directly to another party can be added.
Request for erasure of data	Technical Solution for Right of Erasure [19]	This pattern states what the services must have to provide a simple way of processing the request of erasure of data.
Request for/and update of data	Technical Solution for Update of data [19]	This pattern is a simple solution for what is needed for the data subject to see their data and update any mistakes or changes.
Change of data processing purpose	Negotiation of Privacy Policy [22]	This pattern is already used for another use case but is also relevant because of the opt-in/opt-out options since new purposes can be added and request for new consents.
Consent of minors	Lawful Consent (adapted) [22]	The idea is to adapt the lawful consent pattern but use the guardian of the child for consenting.
Transfer data processing to a third-party	Sticky Policies / Obligation Management [22] Trust Evaluation of Services Sides [22]	The patterns focus on building trust and assuring that the third-party follow the GDPR.

Notify the data subject	Asynchronous notice [22]	One pattern covers possible breaches, and the other covers more general matters.
	Unusual Activities [22]	

4.6. Library

In this chapter, the library will be presented. The use cases and their diagrams will be shown as well as a summary of the patterns and their diagram.

As was said in the chapter above, not all use cases had patterns related to them in the *https://privacypatterns.org*, and [19] didn't have very complete solutions, so two patterns were adapted from that document.

They will be shown to the fullest, as it is a way to show the full template mentioned in section 4.4.

4.6.1. Register in system Use Case

Associated Use Case: Register in system.

To be able to use a service provided by public administration (for example), registration is needed. When registering, the citizen or user (data subject,) provides a set of data to the system, many of which is personal. When storing personal data, consent must be acquired, and the storage must be secure.

The entities present in this use case are:

- Data Subject (Client)
- Data Processor
- [Data Controller]

Associated GDPR principles:

- Purpose limitation
- Data minimization
- Integrity and confidentiality
- The concern about consent

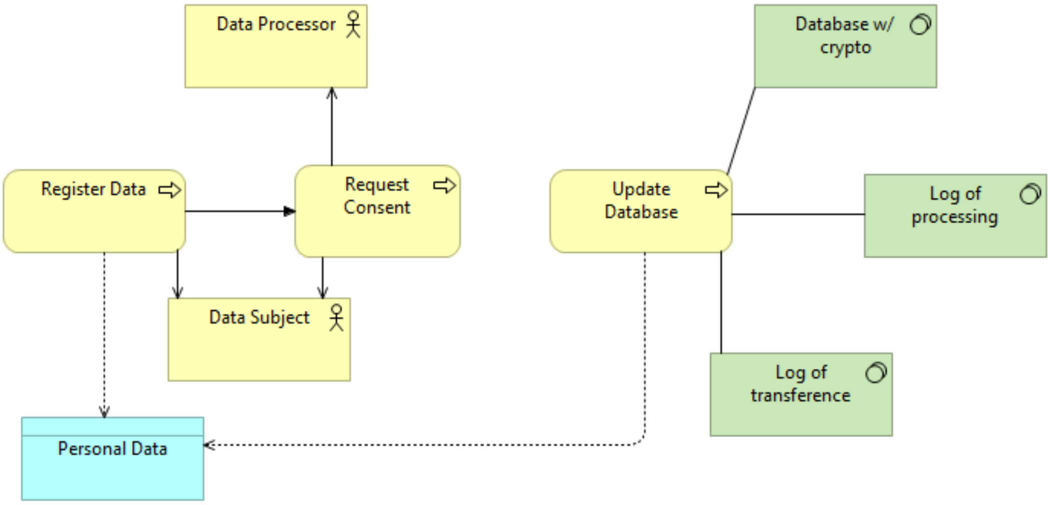
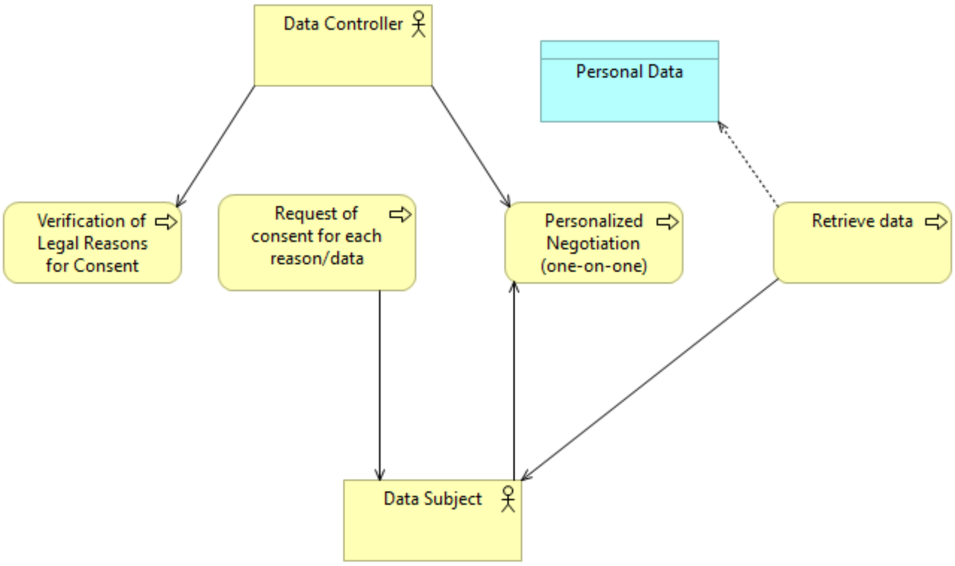


Figure 9 - Diagram of Register in system Use Case

Table 6 - Lawful Consent

Name:	Lawful Consent
Context:	Where data controllers (e.g. organizations) aim to provide a service (or product) to users, there may be opportunities to reuse data, gather feedback, or make use of user data to further their system's value. Many controllers seek to continually collect and utilize this data, often in ways which warrant privacy concerns. For any data processing (including collection), controllers should first obtain consent from the users in question. There are social norms surrounding the use of personal data which need to be adhered to if a controller wishes to avoid scrutiny. Users do not inherently trust controllers to handle their personal data with care for privacy. Without clearly defined boundaries, these users may have justifiable concerns about what is learned about them, and how this information may be used. Additionally, various jurisdictions supply varying compliance requirements, and these controllers need to cater to every market they provide to. Doing otherwise, possibly by disinterest or negligence, may have financial consequences in addition to potential public outcry. Despite this, controllers regularly consider the impact that their decisions may have on competitive edge and resulting profits. The link between better decision making, possibly less sharing, and reduced monetary gains sways some controllers into unlawful forms of consent. Concerned controllers aim to promote trust in any number of ways, potentially including an Awareness Feed and or Privacy Dashboard to properly inform their users. The controller in this context may wish to adhere to the corresponding laws for their users, or above that, genuinely value their users' rights to self-determination.

Problem:	A controller aims to maximize the value of their services by gathering as much sharing and participation as possible, potentially seeing user consent as a barrier to functionality and efficiency. They may inadvertently subvert notions of consent by unnecessarily bundling together desirable services with needs for personal information or downplaying the significance of the data involved. They undermine self-determination at the risk of losing trust from their users and attracting legal investigations which may rule their practices unlawful. *Forces/Concerns:* • Controllers want to encourage participation, and thus may be less concerned with investigating or revealing tradeoffs • Controllers may be tempted to bundle various services under a single broad consent request, pressuring users into agreements they might not otherwise accept • Users often want to make use of new and exciting features, and therefore easily overlook downplayed privacy risks • Some users avoid certain services as they realize the potential privacy risks are not being acknowledged.
Solution:	A user should be given every opportunity to assess their sharing choices prior to making their consent. The controller should aid the user in comprehending the tradeoffs apparent in using each of their services, without over-burdening the user. These consented services should be purposed-separated, so that users may make use of functionality without first granting unnecessary consent. *Rationale:* Controllers need to ensure that anything they do with a user's sensitive or potentially identifying data is legal. This pertains to lawfully obtained consent, for purposes which are clear and lawful in their own right. Additionally, anything they do should be resistant to backlash from users. Implementation: *Separate Purposes:* Services should be separated into distinct processes for which distinct consent is acquired. Each purpose requires its own consent. These permissions need to be given subsequent to ascertaining sufficient awareness in the user about the consequences of that consent. *Freely Given Consent:* The users should not be pressured into providing consent. Instead, the benefits may be presented along with the trade-offs so that the users may make an informed decision. Some users are not necessarily capable of making these

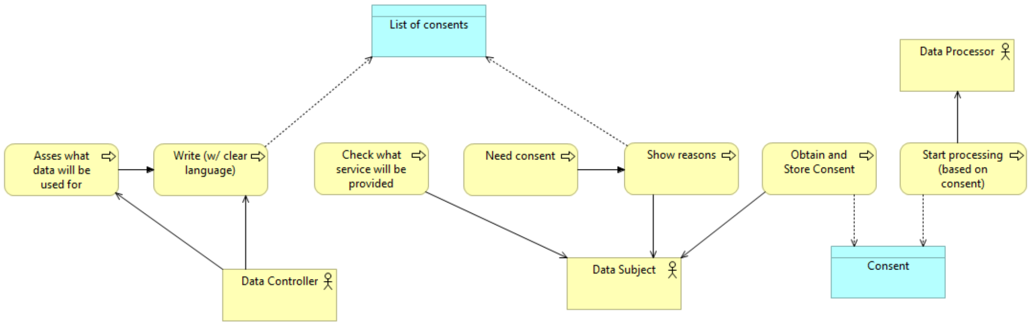
	decisions themselves (e.g. children) and thus provisions need to be made to cater to this. The provided information should not be misleading, as coerced consent is not a valid form of permission. One way to present policies in an accessible manner is through comparative examples (e.g. in addition to further detail, what is unique about our privacy policy?). Providing too much information may also intimidate users into making uninformed decisions, and thus awareness must be garnered in a way which is broadly accessible (see Awareness Feed). Opportunities for further reading should be available, though should not be necessary to understand the trade-offs involved. *Personalized Negotiation:* In more personal services (i.e. one-on-one), personal privacy policies may undergo a formal negotiation. As opposed to user preferences (both at sign-up and through appropriate defaults), understanding a user's personal privacy requirements may benefit from the facilitation of a human representative. This, however, suffers from its own drawbacks where the representative may misunderstand the user's requirements. Even in interpersonal exchanges, controllers should err on the side of caution. Where available, explicit signing of an agreement aids in proving consent.
<u>Solution Diagram:</u>	 Data Controller Personal Data Verification of Legal Reasons for Consent Request of consent for each reason/data Personalized Negotiation (one-on-one) Retrieve data Data Subject *Figure 10 - Diagram of Lawful Consent*
<u>Source:</u>	https://privacypatterns.org/patterns/Lawful-Consent

This pattern focuses on the concern that the controller wants to provide the best service possible to the user (data subject). However, when personal data is needed, they may overlook that consent must be provided for each purpose. Users also want to start using the service right away and overlook privacy risks.

The presented solution starts by making sure the data will be used within legal grounds and lawfully. The controller should show the data subject all the risks for providing the data, and for each purpose, the user’s consent must be requested, and it must be given freely. In some cases, a one-on-one negotiation of what and for what the data will be used could happen. Then, after all the consents are given, the data can be collected.

Table 7 - Informed Implicit Consent

Name:	Informed Implicit Consent
Context:	Processing of user (data subject) information, particularly that which potentially identifies a user or group, requires their explicit informed consent. Inaction is not considered valid consent. However, not all instances make this feasible. As such there are circumstances in which legitimate interests of the controller may justify collection without first obtaining a clear statement of permission to do so. Security footage around a controller's premises, or fraud detection, for example, cannot reasonably be made optional to users of the service (or product). What constitutes legitimate interests in these contexts depends on the relationship and reasonable expectations between the controller and user. As such, sensitive data, or special categories of data, are more difficult to justify.
Problem:	A controller needs to collect and otherwise process reasonable information to fulfill their legitimate interests regarding a user, but cannot feasibly acquire each user's explicit consent. *Forces and Concerns:* • Users should not have to frequently and explicitly consent for regular, everyday, ubiquitous services which are expected and acceptable for legitimate interests • Users do not want to have certain data processed, and need a way to avoid implicitly consenting to it • Controllers do not want to have to obtain explicit consent in real-time bulk for expected and acceptable legitimate interests • Controllers want to ensure that legitimate consent exists before processing
Solution:	Provide clear and concise notice that by using the service, the user implicitly consents to the processing necessary to fulfill legitimate interests. Ensure that this notice is perceived prior to the application of the effects it describes. [Implementation]: Ensure that users are informed sufficiently prior to any processing with clear and concise notice, the complete detail of which should also be accessible. In digital mediums, this is straightforward, working similarly to Cookie Walls on websites. Users should be given the opportunity to choose not to use the service and therefore not be subject to the processing it requires.

	In physical instances it is more difficult to be sure that users take note of this. On devices, lights have often been used to convey a recording state. This, while clear once already subject to processing, is not sufficient however. Instead, large signs are commonplace to indicate the use of data collection. The most familiar example would be "Smile, you're on camera". Of course, this is less clear than "Our premises is recorded for security purposes, by entering you consent to this processing. See more info at [address]". These signs should be posted, visible prior to recording, at all entrances or otherwise where applicable.
Solution Diagram:	 List of consents Data Processor Asses what data will be used for Write (w/ clear language) Check what service will be provided Need consent Show reasons Obtain and Store Consent Start processing (based on consent) Data Controller Data Subject Consent *Figure 11 – Diagram of Informed Implicit Consent*
Source:	https://privacypatterns.org/patterns/Informed-Implicit-Consent.html

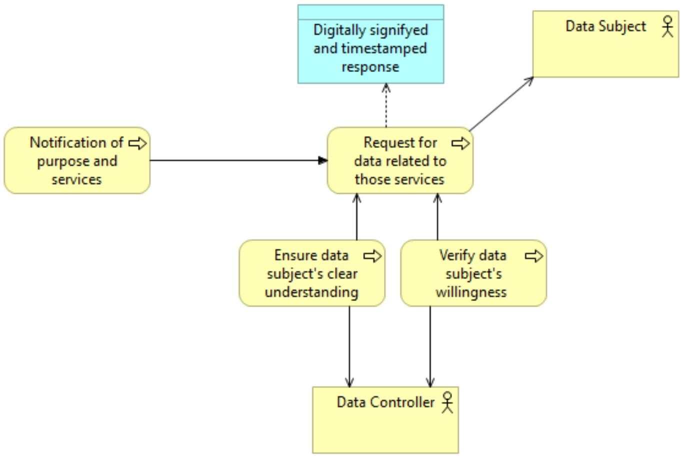
This pattern is concerned with the fact that it may be a hassle to show everything the data will be used for and that all that information may scare the user. Besides, not all purposes for processing need consent, so requesting one may not be necessary. Users may also not want all their data to be processed, and this needs to be considered.

The proposed solution involves assessing the purposes of the possible processings and write clear and concise notices to show to the user. If the services the data subject wants to use requires consent, it should be requested. The data can then be retrieved according to the user's preferences. The notices and the consents should be presented to the user before the usage of the service.

Table 8 - Obtaining Explicit Consent

Name:	Obtaining Explicit Consent
Context:	In order to offer services (or products) to users (data subjects), controllers often need to collect (process) user data. Sometimes this is sensitive, identifying, or just metadata or other information which may be correlated to become more invasive. This nonetheless enables them to offer competitive features and functionality.

	However, controllers are required to obtain unambiguous consent from their users in order to process their personal data in any way. Depending on the legal jurisdiction, there are additional considerations to take into a
Problem:	Controllers which aim to make use of user data, especially that which can be used to identify the user or sensitive aspects about the user, may not do so without a legally binding and sound acquisition of the user's consent. *Forces/Concerns:* • Users want to use services without having to invest an inordinate amount of effort into discovering privacy risks. • Controllers need to be sure that users do not consent out of impatience or intimidation. • Users do not want to consent many times to the same service under the same privacy policy for each and every purpose. • Controllers need to be able to prove that users consented.
Solution:	Provide a clear and concise notification of all pertinent information the service could derive provided it had all the data it asks for. Indicate what this means for features and functionality. Then ask the user whether this tradeoff is something they consent to. If true, digitally signify and timestamp their response, or use Contractual Consent. [Implementation]: The controller must ensure each user's sufficient understanding of the potential consequences. Otherwise the consent might not be informed. They must verify their users' willingness despite those consequences to provide their data for the specific purposes they need. If they do not, the consent might not be freely given. Ensuring that users do not consent based on time constraints, or the intimidation of the information provided, may require testing with a sample. If the sample is representative, it will give the controller a defense against any claims of coercion. The mechanism used for users to signify their consent should be clear. For example, if it is a button, it could read "I consent."

Solution Diagram:	 Digitally signifyed and timestamped response Data Subject Notification of purpose and services Request for data related to those services Ensure data subject's clear understanding Verify data subject's willingness Data Controller *Figure 12 - Diagram of Obtaining Explicit Consent*
Source:	https://privacypatterns.org/patterns/Obtaining-Explicit-Consent

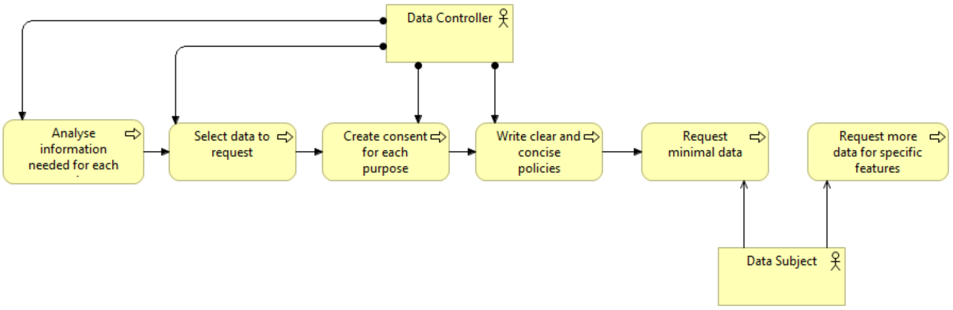
In this pattern, the main concern is that controllers need and want the users' data for their service, and users do not want to be bombarded with privacy risks and consents for the same services and policies. The controllers need to ensure that the data subject consented to the processing and that the consent is not given in impatience.

To solve these matters, the pattern suggests notifying the user of the risks and purposes concisely. Then, when the controller requests for consent, it needs to ensure that the notification is understood and that the consent is given freely and without time constraints; this can be defined by testing earlier with a sample of users. When storing the consent, a digital timestamp should be created with the correspondent consent.

Table 9 - Minimal Information Asymmetry

Name:	Minimal Information Asymmetry
Context:	Users frequently interact with controllers whose services (or products) they have not used before. At this point the knowledge the user has about the controller and its practices, especially regarding privacy, is typically nonexistent. The controller as a whole has a much clearer understanding of its policies. It also begins to know a lot about the user in a short time period, if not already well informed. The user needs to put in sufficient effort to investigate the controller to know about its practices to provide valid consent. The controller needs this valid consent to lawfully process the user's information.

Problem:	Controllers have far more information than the users who utilize their services, which makes the users vulnerable to exploitation. Information asymmetry is generally described as one party having more or better information about a transaction than the other. In order for a healthy consumer relationship to ensue, users should know close to as much about the controller's practices as it would be expected to itself. *Forces and Concerns:* • Users sometimes want to use services of an unknown party, and are cautious about what it might do with their data • Users may not want to provide any more information than necessary, but want the services to function properly • Controllers want users to understand the intentions behind the data they collect, and be content with how they use it • Controllers need to ensure that users understand purposes and means for processing before their consent will be valid
Solution:	Require minimal information from the user, so that only as much personal data as is required, explained, and consented to, is processed. Further reduce the imbalance of policy knowledge by writing clear and concise policies rather than, or in addition to, complex and verbose ones. [Implementation]: Limit the amount of data needed to provide the services necessary to the users, and where appropriate, prefer less sensitive data to do so. Give users the option to opt in to features which require more data, but keep it minimal by default. If the amount of data needed is minimized, then users have less they need to understand, and less to disagree with. This also allows for more simple policies. Making policies more clear and concise is also crucial, as users will not want to sift through long-winded texts to understand what would happen with their data. Highlight important aspects for users themselves, rather than allowing them to become cluttered with legal jargon, detail, and complexity. While certain elements cannot be explained adequately without doing so at length, not all aspects are relevant at once. Some elements may be summarized without the detail, so that users may better understand the current focus. The full detail should still exist however, and be easily located.

Solution Diagram:	 Data Controller Analyse information needed for each Select data to request Create consent for each purpose Write clear and concise policies Request minimal data Request more data for specific features Data Subject *Figure 13 - Diagram of Minimal Information Asymmetry*
Source:	https://privacypatterns.org/patterns/Minimal-Information-Asymmetry.html

When using a new service is expected for users not to know much about it. On the other hand, controllers are much more informed in all the service particularities, so an asymmetry of information knowledge happens. Users may be apprehensive about allowing services to use their data, and controllers want the users to trust them and the service.

The controllers need to analyze which data is, in fact, necessary and only request it. To minimize knowledge asymmetry, clear and concise policies must be formulated for the user to be familiarized with the processing purposes. The pattern also suggests giving the users the possibility to consent to more data for more detailed features.

Table 10 - Awareness Feed

Name:	Awareness Feed
Context:	In a situation where user data is collected or otherwise processed, particularly personal data, many users are concerned about the potential repercussions of their actions. Controllers (e.g.: organizations), which have dynamic and evolving services (or products) which users interact with, may share this concern. This may be for legal, ethical, or public appearance reasons. These controllers also care about the monetary implications of a solution, often including the opportunity cost of informed users against the risks and profits of over-sharing. For-profit organizations regularly want to bolster their market share by overcoming competition with state of the art technologies. These changes may have important consequences, unintentional or otherwise, for users of the system. Controllers want to limit the exposure of these risks to their userbase, even if from a third party, as they are responsible for their data. (…)

Problem:	Users are often unaware of the privacy risks in their data sharing activities, especially risks which are indirect or long-term. How can we best ensure that users become aware of these risks? This problem is agitated by the organizational aim to provide novel and competitive services while keeping users informed. The difficulty of this is frequently underestimated. The pitfalls controllers face as a consequence manifest both in taking shortcuts and in unexpected long-term effects. *Forces/Concerns:* • Users do not necessarily realize the effects of their information sharing, but often want to use new or interesting features • Some users are discouraged from sharing as they do realize that they are not informed about risks to their privacy, but cannot reasonably change that themselves • Controllers aim to provide or utilize novel and or competitive services, but explaining potential risks to privacy in those services is often non-trivial and generates a fear of upsetting the userbase and endangering trust • Some controllers wish to empower users by informing them, but do not want to jeopardize their business model, or ability to process in a timely fashion *Shortcuts:* The appeal of convenience features may sway controllers into flawed implementations which undermine user privacy. Automated decisions, influenced by past actions or by other potentially inaccurate metrics, may result in sharing decisions which users do not approve of. The same holds for features which are not adequately assessed. While a controller might intend all the necessary tools for informed decisions to be present, short-sighted process flows may violate user trust all the same. *Long-term Effects:* Over time, supposedly harmless data may amass into more revealing information, especially when paired with the right metadata. Being able to link user activity to other sources of information may also result in far more exposing situations than expected. Not only are users often unaware of the potential consequences of their actions, even controllers themselves regularly fail to anticipate how revealing their services can be. While some users approach this uncertainty with caution, others will risk their privacy in hopes of using the services. Though the uncertainty might not prevent their participation, it may still jeopardize their trust in the system
Solution:	Warn users about potential consequences before collecting or otherwise processing personal data, early enough to be appreciated and late enough to be relevant. This information should be provided before the point where privacy risks could materialize. If

there is some delay before further processing after collection, the user has some time to review the risks. Until the user accepts them however, that further processing should not take place. (…)

Rationale: It is not likely enough that users are informed prior to being provided a service, nor is it reasonable to expect that consent acquired in bulk is properly informed. Consent is not necessarily freely given, either, if the lack of consent presents a wall to a service that the user wants or believes to need.

A concerted effort needs to be made to present the user with unintimidating information relevant to their privacy risks for a service. Providing too much information lessens the chances that the user will read it, while too little information may not properly inform the user. Informing the user too late also puts the user at unnecessary risk.

By making this effort, the controller avoids accusations of negligence in informing their users.

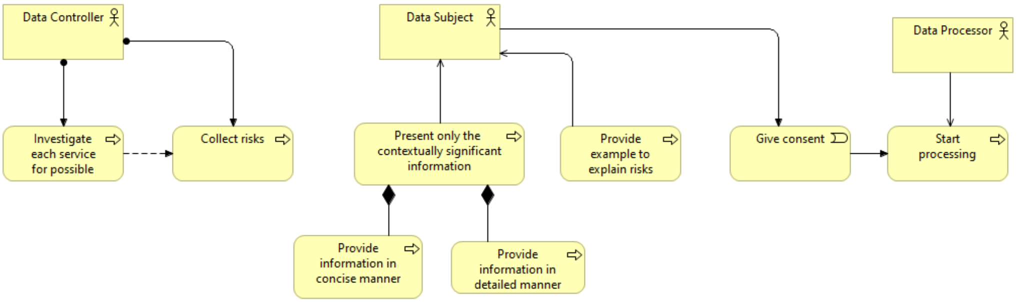
[Implementation]

Every service which makes use of personal information should be investigated by its creators during its creation, or retrospectively if already available. The controller in question is responsible for this. Not only will this affect the user's understanding once presented to them in layperson terms, but it will also allow the controller to realize the privacy impact of their services. This may encourage them to improve the services to be more respecting of privacy. A good solution composes of accessibility, as well as transparency and openness.

Accessibility: There needs to be a balance between the user effort required both to use a service and maintain their privacy. Information about the risks should not be deceptive, or difficult for laypersons to comprehend. Meeting this balance may also be challenging, as fully comprehending the risks involved might require a certain understanding of the system itself.

In order to reduce the quantity of the presented information, only the contextually significant information need be presented. Furthermore, the information should be available in the level of detail sought by the user: in both concise and detailed variants. A short description may be used in Preventing Mistakes or Reducing Their Impact. A more in-depth variation may give them confidence that even if they cannot comprehend it, someone would speak out if something were amiss. In a similar vein, detailed descriptions should be comprehensible enough to avoid accusations of being deliberately complex or misleading.

One way in which to explain the risks involved in a process is through example. This is particularly useful in the case of information aggregation. Visualizing the publicity of data

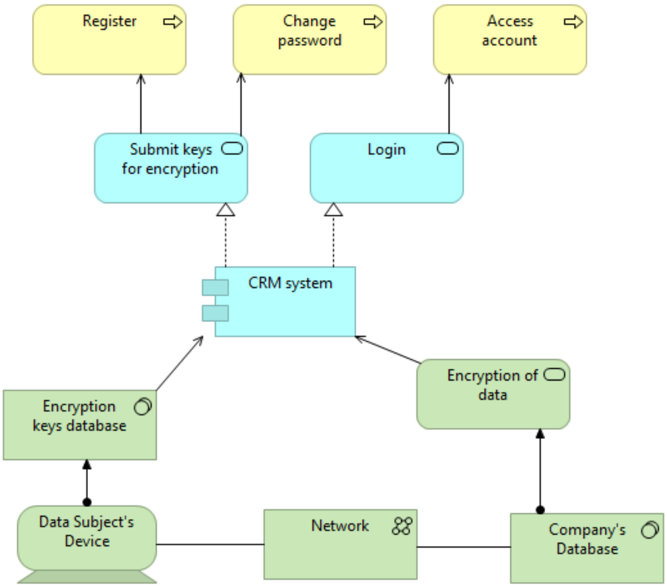
	is also useful, users can see how visible information would be, or is, to the outside world. Similar decisions by those who choose to set examples may also help in influencing informed sharing behaviour. *Transparency and Openness:* Users need to be able to trust that a system does not pose unnecessary risks. Fostering a familiarity with openness and transparency about the processes involved may garner this trust. It allows those who invest time an opportunity to be certain, and those who trust in public perception to be at ease
Solution Diagram:	 Data Controller Data Subject Data Processor Investigate each service for possible Collect risks Present only the contextually significant information Provide example to explain risks Give consent Start processing Provide information in concise manner Provide information in detailed manner *Figure 14 - Diagram of Awareness Feed*
Source:	https://privacypatterns.org/patterns/Awareness-Feed

Many users are not aware of the risks of sharing their data with a service. Others care about the risks but do not fully understand the repercussions because of the controllers' insufficient explanation. The organizations also care about the safety and awareness of the users as well as the trust they have in their services.

To solve this, it is essential for the controller to analyze and create a collection of the possible risks (like a Privacy Impact Analysis, PIA), then these risks must be shown to the user. Only the important and relevant information should be presented to the user to prevent an overflow of information. The service can provide examples for a better comprehension of the risks. Then, only after the data subject gives consent, the processing can begin. The idea is to create an accessible, open, and transparent service that allows the user to understand the risks and trust the service.

Table 11 - Encryption with user-managed keys

Name:	Encryption with user-managed keys
Context:	User wants to store or transfer their personal data through an online service and they want to protect their privacy, and specifically the confidentiality of their personal information. Risks of unauthorized access may include the online service provider itself, or third parties such as its partners for example for backup, or government surveillance depending on the geographies the data is stored in or transferred through.

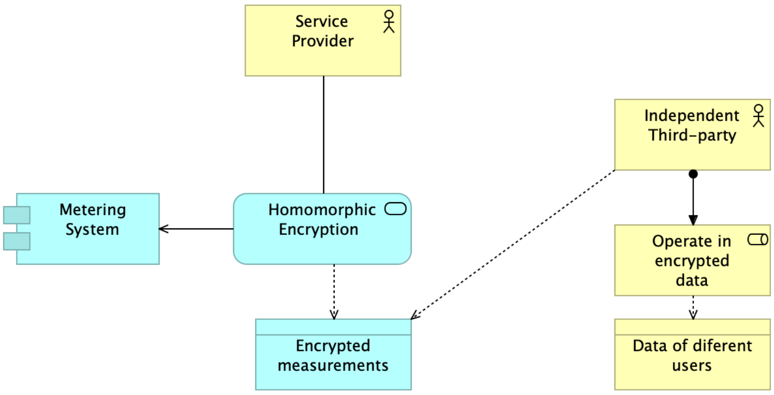
Problem:	How can a user store or transfer their personal information through an online service while ensuring their privacy and specifically preventing unauthorized access to their personal information? Requiring the user to do encryption key management may annoy or confuse them and they may revert to either no encryption, or encryption with the online service provider managing the encryption key (affording no protection from the specific online service provider managing the key), picking an encryption key that is weak, reused, written down and so forth. Some metadata may need to remain unencrypted to support the online service provider or 3rd party functions, for example file names for cloud storage, or routing information for transfer applications, exposing the metadata to risks of unauthorized access, server side indexing for searching, or de-duplication. If the service provider has written the client side software that does the client side encryption with a user-managed encryption key, there can be additional concerns regarding whether the client software is secure or tampered with in ways that can compromise privacy.
Solution:	Encryption of the personal information of the user prior to storing it with, or transferring it through an online service. In this solution the user shall generate a strong encryption key and manage it themselves, specifically keeping it private and unknown to the untrusted online service or 3rd parties.
Solution Diagram:	 Register Change password Access account Submit keys for encryption Login CRM system Encryption keys database Encryption of data Data Subject's Device Network Company's Database *Figure 15 - Diagram of Encryption with user-managed keys*
Source:	https://privacypatterns.org/patterns/Encryption-user-managed-keys

Users want their data protected and not accessed by unauthorized people when transferring it to an online service. Suppose the controller requires the data subject to use encryption. This may confuse them, and they may choose not to encrypt their personal information, making it more vulnerable to privacy breaches.

This pattern's solution is to encrypt the data before it is stored and/or transferred to another party. The encryption keys are created and managed by the user. The creation may happen when they register in the service, for example.

Table 12 - Aggregation Gateway

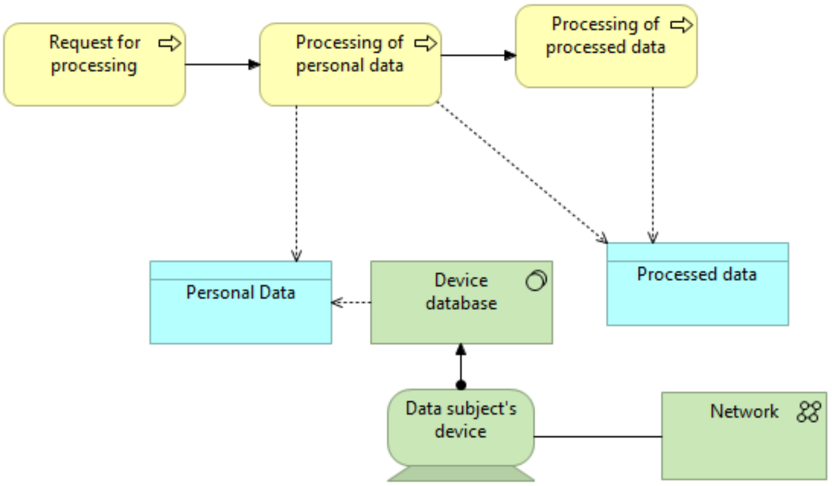
Name:	Aggregation Gateway
Context:	A service provider gets continuous measurements of a service attribute linked to a set of individual service users.
Problem:	The provision of a service may require detailed measurements of a service attribute linked to a data subject to adapt the service operation at each moment according to the demand load. However, these measurements may reveal further information (e.g. personal habits, etc.) when repeated over time
Solution:	A homomorphic encryption (e.g. Paillier) is applied at the metering system, using a secret shared with the service provider (generated by applying e.g. Shamir's Secret Sharing Scheme). The encrypted measurements from a group of users are transmitted to an independent yet trusted third party. This third-party cannot know about the content of each measurement (as it is encrypted), but it can still operate on that data in an encrypted form (as the encryption system is homomorphic). There are different trusted third parties for each group of users. In order to improve the privacy resilience, each user may belong to several groups at the same time. The trusted third-party aggregates the measurements from all the users in the same group, without accessing the data in the clear at any time. The service provider receives the encrypted, aggregated measurement and decrypts it with the shared secret. A feeder metering system can be added as a measuring rod which introduces a comparison for each group of meters. Let the service provider have reliable access to the aggregated load at every moment, so as to fulfil its operating requirements, without letting it access the individual load required from each specific service user.

Solution Diagram:	 Service Provider Independent Third-party Metering System Homomorphic Encryption Operate in encrypted data Encrypted measurements Data of diferent users *Figure 16 - Diagram of Aggregation Gateway*
Source:	https://privacypatterns.org/patterns/Aggregation-gateway

This pattern depends on thrusted third parties to operate in the encrypted data of different users. The data is encrypted and then distributed to several parties, without them being aware of whose data they are processing.

Table 13 - User data confinement pattern

Name:	User data confinement pattern
Context:	This pattern may be used whenever the collection of personal data with one specific and legitimate purpose still pose a relevant level of threat to the users' privacy.
Problem:	The engineering process is biased to develop system-centric architectures where the data is collected and processed in single central entities, forcing users to trust them and share potentially sensible personal data.
Solution:	The solution is to shift the trust relationship, meaning that instead of having the customer trust the service provide to protect its personal data, the service provider now haves to trust the customers' processing. In the smart meter example, the smart meter would receive the monthly tariff and calculate the customer's bill which will be then sent to the energy provider where it will be processed. The main benefit is that at no moment the personal data has left the users trusted environment. Avoid the need for trust in service providers and the collection of personal data.

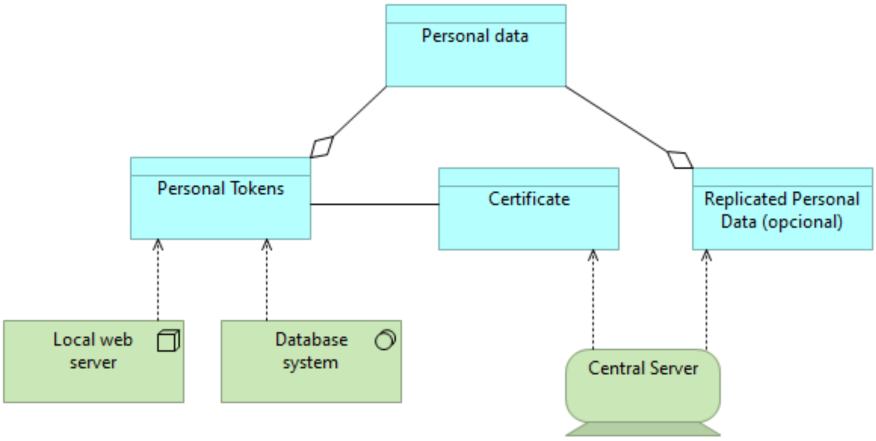
Solution Diagram:	 Request for processing Processing of personal data Processing of processed data Personal Data Device database Processed data Data subject's device Network *Figure 17 - Diagram of User data confinement pattern*
Source:	https://privacypatterns.org/patterns/User-data-confinement-pattern

Usually, the data is stored and processed by the systems of the companies that provide the service. This forces the user to trust the service they are using one-sidedly.

The solution is to change this dynamic and make the service provider trust the users to store their data. The data is stored in the user's device, and the service only receives the data that is needed to a particular functionality process it in a general way, and only then the processed data (that no longer identifies the data subject) is sent and processed on other parties.

Table 14 - Personal Data Store

Name:	Personal Data Store
Context:	The pattern is applicable to any data produced by the data subject (or originally under his control) as opposed to data about him produced by third parties.
Problem:	Data subjects actually lose control over their data when they are stored on a server operated by a third party.
Solution:	A solution consists in combining a central server and secure personal tokens. Personal tokens, which can take the form of USB keys, embed a database system, a local web server and a certificate for their authentication by the central server. Data subjects can decide on the status of their data and, depending on their level of sensitivity, choose to record them exclusively on their personal token or to have them replicated on the central server. Replication on the central server is useful to enhance sustainability and to allow designated third parties (e.g. health professionals) to get access to the data. Enhance the control of the subjects on their personal data.

Solution Diagram:	 Personal data Personal Tokens Certificate Replicated Personal Data (opcional) Local web server Database system Central Server *Figure 18 – Diagram of Personal Data Store*
Source:	https://privacypatterns.org/patterns/Personal-data-store

This pattern concerns personal data, not data related to the data subject created by the system, and how users may lose control of their data when it is transferred to third parties.

The solution is to have a central server and personal tokens that work like keys to authenticate certificates. The data is stored on the user's personal token, and if it is requested, it may be replicated in the central server, making this the only way the company can access the data.

4.6.2. Inform of Breach Use Case

Associated Use Case: Inform of Breach

The GDPR articles 33 and 34 describe what to do when a breach occurs. This can be divided into two parts (like the two articles). Firstly, the processor must inform the controller, who must notify the supervisory authority of the breach within 72 hours. This notification has to follow the guidelines present in Art 33(3). Furthermore, if the breach risks the data subject's rights, they must also be informed, except in the situations stated in Art34(3).

The entities present in this use case are:

- Data Subject (Client)
- Data Processor
- Data Controller

Associated GDPR principles:

- Integrity and confidentiality
- Principles of lawfulness, fairness, and transparency

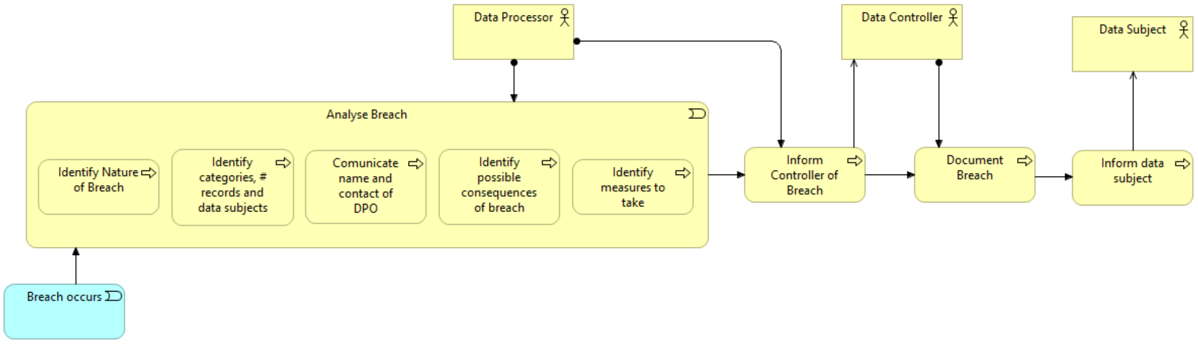
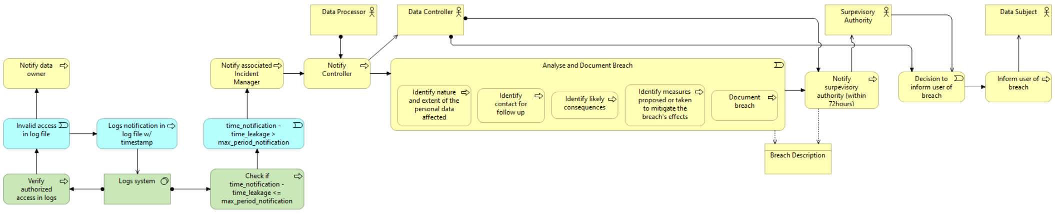

Data Processor
Data Controller
Data Subject
Analyse Breach
Identify Nature of Breach
Identify categories, # records and data subjects
Comunicate name and contact of DPO
Identify possible consequences of breach
Identify measures to take
Inform Controller of Breach
Document Breach
Inform data subject
Breach occurs

Figure 19 - Diagram of Inform of Breach Use Case

Table 15 - Data Breach Notification Pattern

Name:	Data Breach Notification Pattern
Context:	Controllers of services (or products) provided to users collect mass amounts of data, a lot of it personal, to improve the quality and user experience of that service. This is all to be done under the informed consent of the user, who should properly understand the risks involved for their data. One such risk is that of unauthorized access, modification, removal, or sharing of data. If such a data breach occurs, notification is required. Any controller within (or providing services or products within) the EU must notify the supervisory authority of their main establishment or representative. This must occur within 72 hours unless justified. Notifying users is dependent on whether they are sufficiently affected.
Problem:	When data breaches occur, numerous risks become apparent for multiple parties, these parties need to be notified and the risks need to be mitigated. Subsequent instances should be prevented through lessons learned. *Forces and Concerns:* • Users want to know if anything has happened to compromise their data, their security, or their privacy • Users want the controller to mitigate the risks before and after a breach to the best of their ability • Controllers want to prevent risks from materializing and place measures against breaches happening in future Controllers also want to prevent users from suffering consequences from the breach, or from ignorance of the breach.
Solution:	Detect and react to data breaches quickly, notifying the supervisory authority of details, particularly risk mitigation, in order to establish whether users must also be informed. Properly handling these events will strengthen user trust rather than weaken it.

	[Implementation] A monitoring system logs access to [personal data] along with a time-stamp. A notification process continuously verifies that only authorized access is listed in this log file, and in case of unauthorized access notifies the data owner and logs the notification action in the log file, again accompanied by a time-stamp. A notification monitoring process finally continuously checks that t_n - t_l <= max_np (t_n denoting the time of notification, t_l the time of data leakage, max_np the maximally allowed period of notification). In case t_n - t_l > max_np it alerts the [associated] Incident Manager. In the event of a breach, the controller should first notify the supervisory authority within 72 hours of it's discovery, and no later without sufficient justification. The processor of personal data, where not also the controller, should notify the controller immediately. Notification to the authority should include the nature and extent of the personal data affected, the contact for follow up, likely consequences, and the measures proposed or taken to mitigate the breach's effects. If absolutely necessary these details can be provided as they become available. Any breaches should also be documented for future review. Where users are affected in a manner which risks their personal rights and freedoms, they shall also be informed of at least the contact, consequences, and measures to be taken, without undue delay. This is not the case if disproportionate effort would be needed, the data remains protected, or the risk is already sufficiently mitigated. The supervisory authority shall assist in determining whether informing users is necessary. Note that associations or other representative bodies may prepare codes of conduct for data breach notifications. These notifications may also be affected by binding corporate rules, or guidelines, recommendations, and best practices from the board, to promote consistency
<u>Solution Diagram:</u>	 Data Processor Data Controller Surpevisory Authority Data Subject Notify data owner Notify associated Incident Manager Notify Controller Analyse and Document Breach Identify nature and extent of the personal data affected Identify contact for follow up Identify likely consequences Identify measures proposed or taken to mitigate the breach's effects Document breach Notify surpevisory authority (within 72hours) Decision to inform user of breach Inform user of breach Invalid access in log file Logs notification in log file w/ timestamp time_notification - time_leakage > max_period_notification Breach Description Verify authorized access in logs Logs system Check if time_notification - time_leakage <= max_period_notification *Figure 20 - Diagram of Data Breach Notification Pattern*
<u>Source:</u>	https://privacypatterns.org/patterns/Data-breach-notification-pattern

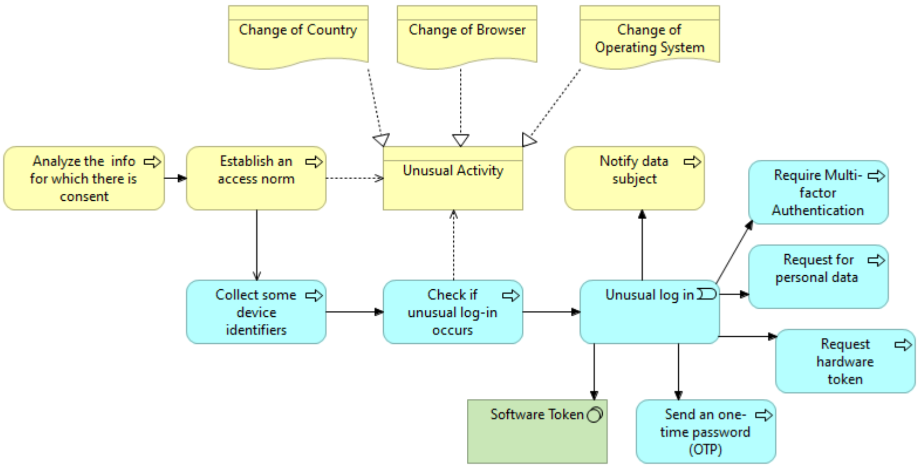
When a breach occurs, many parties need to be notified, and the risks need to be addressed and resolved. Users want to know when something happens to their data and want the breaches to be solved. The controllers want to minimize the risks and new possibilities of breaches and want the users to be informed on these matters.

The solution consists of having a log system that monitors the accesses of the data. In the event of a breach, the processor should notify the controller, and the controller must notify the authorities with a list of information related to the breach. If the breach affects the data subject, they should also be informed of the occurrence. This pattern follows particularly well the articles of the GDPR that deal with the matter of data breaches.

Table 16 - Unusual Activities

Name:	Unusual Activities
Context:	Services (or products), particularly over the Internet, tend to use username and password-based authentication. This security mechanism proves most convenient for users, as it is commonplace and simple compared to the more secure alternatives. It is also subject to common shortcomings, however. Passwords become less secure the longer they remain unchanged, are often vulnerable to brute force, snooping, and phishing attacks, and cannot be proven to be held solely by the user. This complicates the certainty of the authentication, and thereby the authenticity of any decision made by the user, including consent. Controllers may also derive additional factors, however, such as device or access specific information. If location is provided, for example, it may hint at unlikely account activity.
Problem:	Username and password authentication alone has varying reliability for proving decisions taken by a user, especially when concerning more sensitive actions. Controllers need to enhance their certainty that any consent provided is legitimate. *Forces and Concerns:* • Users want to be able to authenticate easily and quickly, but also do not want controllers to accept decisions made by intruders • Users want to know that their password is compromised, so that they can change it, especially if they use derivatives elsewhere • Controllers want to protect user accounts from unauthorized access • Controllers do not want to allow actions which the user did not truly consent to. A balance should be made between the insecurity of username and password authentication and the inconvenience of multi-factor authentication. If measures affect usability or privacy too greatly, users will stop using the system. While the rate of false positives must not be too high, they are far preferable to undetected intrusions. Facebook, for example, makes use of its resource of friendship and photos. Their decision is based on the assumption that it is very unlikely for a hacker to recognize the friends. Actually the assumption may not hold true in some scenarios, because many of the photos are public and can be viewed under another account, or can be

	identified with the help from a large-scale tagged photo collection and machine learning. Persuading the user into carrying a hardware token everywhere only for occasional multi-factor authentication may be difficult, but it might worth the effort for financial services
Solution:	Analyze the available information for which there is consent to establish an access norm. Test this against future access to identify unusual activities. When this occurs, alert the user and use multi-factor authentication while re-establishing certainty. The authenticated user should be able to review and take further action. [Implementation] Typically, a sign-in to a website is in the form of an HTTP request, which contains many customized settings of the browser, including the type of the browser and operating system as well as the architecture (User-Agent header), the Cookie (Cookie header), language preferences (Accept-Languages header). Apart from these, the website can get the IP address of the user, which may be mapped to a certain country/area through GeoIP. [These] can be used to tell if a browser is 'new' to the website. The website can have its rules to determine if an access is 'suspicious', for example, an access from a new country / browser / operating system is considered suspicious. By running native code, the application can [consensually] collect some [device identifiers], including the operating system environment settings (e.g. the list of running processes), the hardware parameters (such as the ID of the CPU), and device UUIDs (provided by mobile operating systems like iOS). By completing a network request, the service also retrieves the IP address of the [device]. [These] can be used to tell if a [device] is 'new' to the service. The service can have its rules to determine if a sign-in is 'suspicious', for example, an access from a new country / [device] / operating system is considered suspicious. *Require Multi-factor Authentication:* In case of a suspicious [activity], multi-factor authentication may be a way to let the legitimate user in. The service can request [further authentication], such as: *A software token:* Examples include Google Authenticator which runs on mobile phones and implements RFC6238 TOTP security tokens. *A hardware token (disconnected):* Examples include a token issued by a bank which displays digits, which is similar to a software token. *A hardware token (connected):* The token may exchange a longer secondary password than the previous one, which means it's safer.

	Personal data like date of birth, [or civil identification]: Obviously not a good choice here because it cannot be changed. *An one-time password (OTP) sent to the registered E-mail address / mobile phone:* Depending [on] the type of the service, [the user may use] the same password for the E-mail address, or [may lose their mobile phone]. Using multi-factor authentication only in case of suspicious [activity] is more convenient [than] using it all the time, but is less secure. *Notify Account Holders of Unusual Activities:* When a suspicious sign-in is detected, it may be a sign that the password has already been leaked. Depending on the type of the service, it can notify the user about the suspicious sign-in through E-mail, telephone, or other means. Here the immediate notification can also be used in the multi-factor authentication. For services that can be logged on from multiple devices at the same time, the user should be able to check the existence of other sessions, and review recent [activity].
Solution Diagram:	 Change of Country Change of Browser Change of Operating System Analyze the info for which there is consent Establish an access norm Unusual Activity Notify data subject Require Multi-factor Authentication Request for personal data Collect some device identifiers Check if unusual log-in occurs Unusual log in Request hardware token Software Token Send an one-time password (OTP) *Figure 21 - Diagram of Unusual Activities*
Source:	https://privacypatterns.org/patterns/Unusual-activities

Many services use usernames and passwords for authentication since it is easy and straightforward for the users to work with. This way is not very reliable, and malicious access can happen. A way to prevent this is by using multi-factor authentication, but it can become a hassle for some users. Controllers want the users' trust and the safety of their data, and users want to be able to prevent or stop unauthorized access by changing their password, for example.

The solution passes by establishing an access norm and monitor the access to the account and data. When unusual activity is detected, the user is notified and is requested another form of authentication. This way, if it is the user accessing from a different device, for example, it can continue with the authentication; if not, they can stop the intruder and change the password to prevent more un-authorized authentications.

4.6.3. Request for restriction on personal data's processing Use Case

Associated Use Case: Request for restriction on personal data's processing

Art. 18 of the GDPR is about the right to restriction of the processing. This article states that if the data subject suspects the accuracy of the stored data or the unlawfulness of the processing, the subject can request for restriction of processing, not just the erasure of its data. This restriction may be granted to these suspicious and other grounds in Art.18 (1). It is important to notice that if the subject presents no cause, the request is not arranged.

The entities present in this use case are:

- Data Subject (Client)
- Data Controller

Associated GDPR principles:

- Trueness and accuracy
- Principles of lawfulness, fairness and transparency
- Data minimization
- Data subject's right to restriction of process.

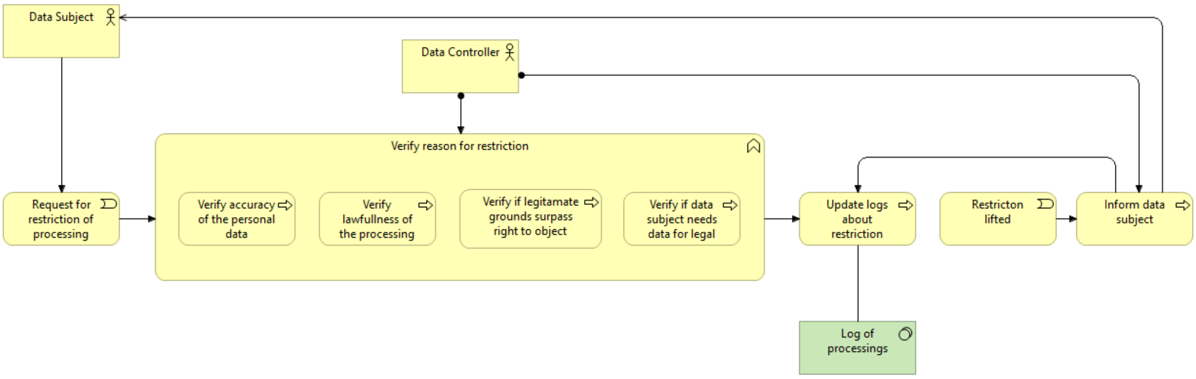
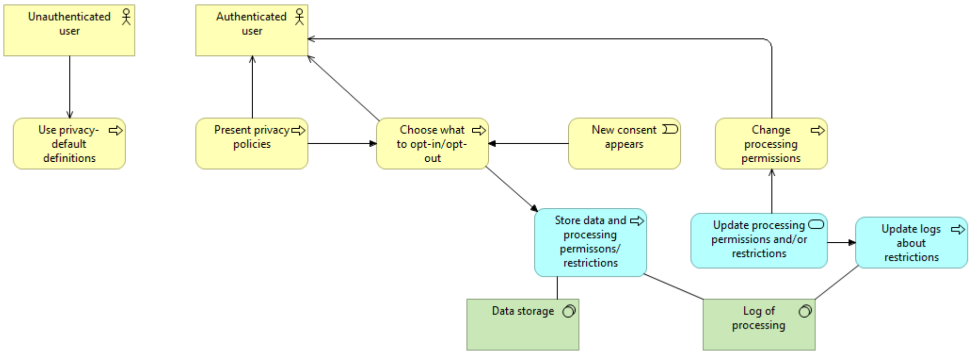

Data Subject
Data Controller
Verify reason for restriction
Request for restriction of processing
Verify accuracy of the personal data
Verify lawfullness of the processing
Verify if legitamate grounds surpass right to object
Verify if data subject needs data for legal
Update logs about restriction
Restricton lifted
Inform data subject
Log of processings

Figure 22 - Diagram of Request for restriction on personal data's processing use case

Table 17 - Negotiation of Privacy Policy

Name:	Negotiation of Privacy Policy
Context:	Often when users find a service (or product) they would like to use, and begin signing-up, they are immediately exposed to assumptions which may not hold for them. As users have differing privacy priorities, a controller cannot guess as to what settings best accommodate them. Since these preferences may be intricate, users cannot be expected to specify them in detail all at once or before using the service.
Problem:	Users have sometimes wildly different priorities regarding their privacy, though a controller does not know these details when a user first joins a service. There is a temptation to provide these users the settings the average user uses. *Forces/Concerns:* • Users are different and do not all fall under one universal setting without some being unsatisfied. • The controller wants to cater to user individuality. • Getting users to specify all of their individual tastes before using a service will make some users abandon the process. Some settings may be missed, and many users will be upset
Solution:	As users begin to use a service, determine their individual privacy sensitivities by allowing them to opt-in/opt-out of account details, targeted services, and telemetry. When a user's preference is not known, assume the most privacy-preserving settings. It should always take more effort to over-share than to under-share. [Implementation] Unauthenticated users should enjoy the most privacy-preserving defaults. When a user joins the service, they may be presented with [excerpts or summaries of] a privacy policy, which they can use to inform their choices. Using simple, recognizable controls, users can be asked to opt-in (for explained benefits) or opt-out (at explained costs) before any of their data is used. They can then be asked for additional consents further down the line as they become contextually relevant. In this way, only the needed consent is asked for as the controller's understanding of the user's preferences improves. This can allow the service to determine which solicitations users are individually likely to consider, and which ones will only waste their time or upset them.

Solution Diagram:	 Unauthenticated user Authenticated user Use privacy-default definitions Present privacy policies Choose what to opt-in/opt-out New consent appears Change processing permissions Store data and processing permissons/ restrictions Update processing permissions and/or restrictions Update logs about restrictions Data storage Log of processing *Figure 23 - Diagram of Negotiation of Privacy Policy*
Source:	https://privacypatterns.org/patterns/Negotiation-of-Privacy-Policy

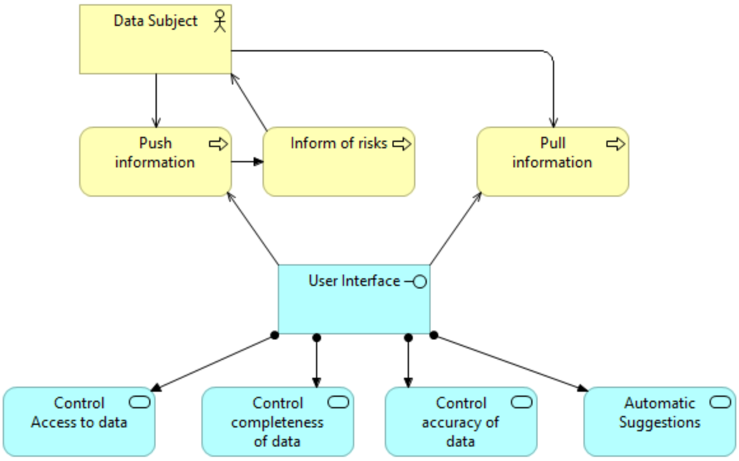
The controllers want the services to accommodate the user's needs and match their privacy priorities, but this is very hard to achieve. If predefined settings are used, many users may not feel safe, and some may feel smothered. If the users must choose settings for every matter, it may become bothersome and can make them give up on the service.

The proposal is allowing users to change their privacy settings using opt-in/opt-out selections. They can start with a level of security and change that level, informed with an explanation of the costs throughout the service usage. The pattern also suggests that users that do not have an account to use the most privacy-preserving default since the changes they want cannot be stored.

Table 18 - Reasonable Level of Control

Name:	Reasonable Level of Control
Context:	Users have certain expectations about what level of privacy they can expect in certain contexts. In general, they are given the means to provide themselves with as much or little shielding from intrusions as they need. This expectation carries over to usage of services (or products) offered by a Controller. Users expect that they can have an impact on what about them is known to a service, or others that use the service.
Problem:	Users expect to be afforded sufficient self-determination over what information about them is collected or otherwise processed. The level of information and control desired, however, varies from person to person, as does the negative response when expectations are not met. *Forces/Concerns:* • Users want to share and be shared with, but have varying limits on what they feel comfortable sharing. • They have their own conceptions on what is worth withholding, and different regards for information sensitivities.

	- Not all users trust a service to handle their information with the same care they feel is due. - Many users want others to be able to know certain things about them on request, sometimes even in real-time.
Solution:	Allow users to selectively and granularly provide information to a service, or its users, and have select information available to user-defined or predetermined groups. [Structure] Users should be able to push their chosen information to (or have it pulled by) those they grant access. Using push mechanisms, users will have the greatest level of control due to the fact that they can decide the privacy level of their data case by case. Pull mechanisms are less granular, as granting access to a group or individual continues until that access is denied. Within this time frame, the sensitivity of the data may fluctuate. However, the user should have the ability to retract access at will, and thus, can manage their own identified risks. Users should also be made aware of the potential risks of over-sharing and increased sensitivity of data over time. This creates a complementing relationship between many Inform patterns, including Ambient/Asynchronous Notice, Preventing mistakes or reducing their impact, as well as Awareness Feed, Privacy Dashboard and their compounded patterns. [Implementation] When users are pushing their information to a service, design the user interface such that where appropriate, controls define the access, granularity, completeness, accuracy, etc. of the information being shared. Elsewhere, ensure that any required fields are truly required, and that the completeness needed for those fields be indicated. When there are automatic suggestions, let users redefine or remove the information before it is collected by the service. These automatic suggestions should also not take place without consent. Where information is provided on a continual basis to those granted access, provide the user with the necessary tools. They should be able to indicate who falls within a group, and what exactly that group can access, for how long, at what granularity, how far back they can look, and so forth

Solution Diagram:	 Data Subject Push information Inform of risks Pull information User Interface Control Access to data Control completeness of data Control accuracy of data Automatic Suggestions *Figure 24 - Diagram of Reasonable Level of Control*
Source:	https://privacypatterns.org/patterns/Reasonable-Level-of-Control

The concern here is that users may want to share their data but have different limits on what they are comfortable sharing. Some users have more trust in the service, while others have different views of what is safe to share and what is not. Controllers, of course, want the users' trust and to be able to work within the limits of comfort.

The solution is to have a push and pull system that allows users to push the information they want to share, informing them of the risks, and pull the information they do not want or no longer feel comfortable sharing. The key is design an interface that allows this dynamic without affecting the completeness and accuracy of the data the service requires, as well as providing automatic suggestions for users who may not want, at first, to think much about these matters.

4.6.4. Request of personal data Use Case

Associated Use Case: Request of personal data

Art. 15 specifies that the data subject has the right to request for his/her personal data and additional information related to the processing Art.15(1)) if it is confirmed that their data is being processed. A copy of the data undergoing processing shall be provided, and any further copies requested may have a reasonable fee associated. If the request is made by electronic means, and no constraint is presented, the information shall be provided in a commonly used electronic form (for example, an excel).

The entities present in this use case are:

- Data Subject (Client)
- Data Controller

Associated GDPR principles:

- Principles of lawfulness, fairness, and transparency
- Data subject's right of access.

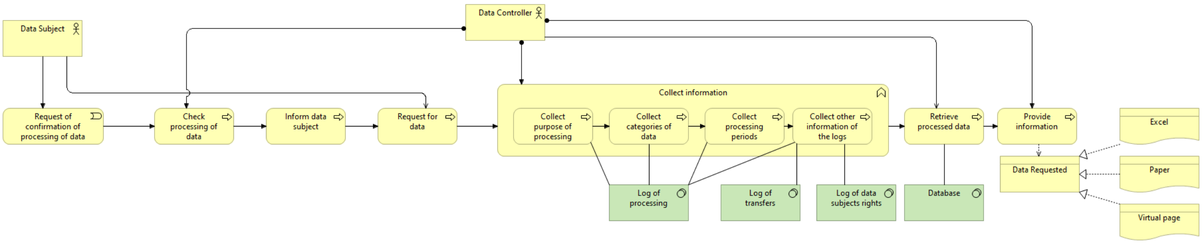
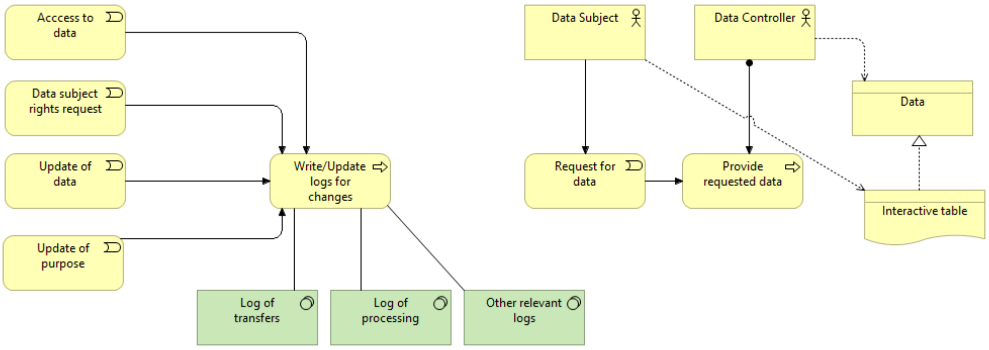


Figure 25 - Diagram of Request of personal data use case

Table 19 - Personal Data Table

Name:	Personal Data Table
Context:	Controllers which maintain software systems that process user data, especially identifying or sensitive data, are subject to various laws. In the case of personal data, transparency about processing is particularly important. Users (the data subjects) also care to know about what data is used, and what might be done with that data, at various degrees. Users do not often want to be constantly notified or reminded, as many of them would rather spend their time actually using the system. Some users, however, care about more intricate detail, and are entitled to it. Nonetheless, if verbose information is provided, it should be sensible.
Problem:	The controller wants to be upfront about what they know and can do with personal data which might be of importance to those users. They only want users to know about data and risks pertaining to them specifically. *Forces and Concerns:* • The controller wants to show the actual data they process, as well as what they do with it, as opposed to just describing policy • Users want full transparency, with detailed explanation as well as easily and quickly understood overviews • Controllers do not want this transparency to ruin trust, but to strengthen it • The controller wants to keep the data on their servers, while still allowing users to automatically view their own data.
Solution:	Keep track of the processing that occurs on personal data so that users can view the activities associated with their data and review their preferences in a tabular environment. [Structure] Which information A table that shows the overview. The overview could show: − Which data − Why collected − How used/for which purpose collected − Who has access to the data − Who the user authorized for access − Which consent the user

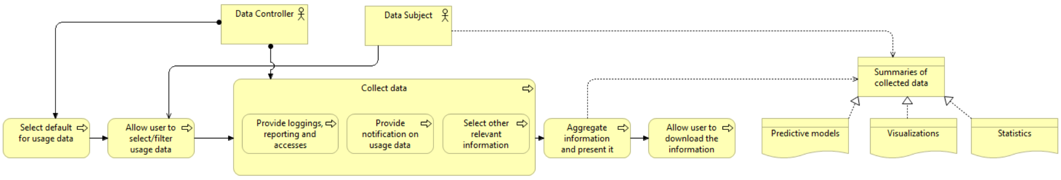
	has given for specific data – To which parties the data is disclosed – Who has seen the data – Whether the data can be hidden – Whether the data can be removed – How long the data is stored – How datasets are combined to create richer (privacy sensitive) information. Note that this may violate local laws and regulations – With which other information the data is combined Where in the application flow Options are (not mutually exclusive): – At the service's help section – At the service's privacy section – Through a separate menu item – At a myData section of the service Amount of information A table can show a lot of information or can be adjustable by the user to tweak which information to show, and which values (e.g. which range). From the table links to applicable other pages/screens can be given, to allow a user to easily change privacy settings (or possibly delete data) or visit websites of data buyers. A way to present more detail than visible at the overview table is to apply the Overview beside detail user interface pattern (Laakso 2003). [Implementation] Provide users with access to an interface which displays their data in useful dataset views, and give them the option for raw information. See the following table for an example. |Type of Data|Data|Date Recorded|Accessed by| |--|--|--|--| |data type a|data itself|date a|person one| |data type b|data itself|date b|person one, person two| To be really transparent, also show things like how and why data was used, who of your organization has access to the user's personal data, what was downloaded or sent to a specific third party, and when all these events happened. The table can present all the data at once, or order it in categories, that may be further detailed when the user selects a category**.**
Solution Diagram:	 Acccess to data Data subject rights request Update of data Update of purpose Write/Update logs for changes Log of transfers Log of processing Other relevant logs Data Subject Data Controller Data Request for data Provide requested data Interactive table *Figure 26 - Diagram of Personal Data Table*
Source:	https://privacypatterns.org/patterns/Personal-Data-Table

This pattern focuses on the transparency of processing and the laws that come when processing personal data, and the users' right to know what is happening. The controller must comply with the regulations and provide a transparent service. They want the users' trust to enhance by providing information on the processing with clear and straightforward language.

The first step is to keep some logs system that stores all the processing performed on the data. The pattern even provides an example of how this information can be stored and suggests storing the why and how the data is being used. When requested, this table can be presented to its fullest or ordered by categories, according to the user's needs.

Table 20 - Privacy Dashboard

Name:	Privacy Dashboard
Context:	A service (or product) which processes personal data of users may make that data accessible to them. This is often the case whether conforming to laws about self-determination and notice, or merely wanting to provide an additional privacy consideration for the sake of users. The controller concerned wants to open up about the data they have processed, and to improve the ease of use for configuring privacy settings**.**
Problem:	A system should succinctly and effectively communicate the kind and extent of potentially disparate data that has been processed. Users may not remember or realize what data a particular service or company has collected, and thus can't be confident that a service isn't collecting too much data. Users who aren't regularly and consistently made aware of what data a service has collected may be surprised or upset when they hear about the service's data collection practices in some other context. Without visibility into the actual data collected, users may not fully understand the abstract description of what types of data are collected; simultaneously, users may easily be overwhelmed by access to raw data without a good understanding of what that data means. *Forces and Concerns:* • Controllers want to provide users with sufficient information to determine how it is used, and to prevent regrettable sharing decisions • Controllers want to prevent both over and under-sharing, so as to provide users with the best experience possible • Users often do not realize the privacy risks in providing their personal data • Users do not want to be subjected to too many or overly detailed notifications, as they will quickly make a habit of overlooking them.

Solution:	Provide successive summaries of collected or otherwise processed personal data for a particular user, representing this data in a meaningful way. This can be through demonstrative examples, predictive models, visualizations, or statistics. *Where users have choices for deletion or correction of stored data, a dashboard view of collected data is an appropriate place for these controls (which users may be inspired to use on realizing the extent of their collected data).* [Structure] A variation of the privacy dashboard Privacy Mirrors focuses on history, feedback, awareness, accountability, and change. [Implementation] Implementing this pattern is a matter of providing logging, reporting, and other informational access and notifications on user-selected/filtered, appropriately defaulted, relevant usage data. Aspects which the controller wishes to inform their users about may include the collection and aggregation of their data, particularly personal data which: – changes over time, – is [processed] in ways that might be unexpected, – is invisible or easily forgotten, or – is subject to correction and deletion by users.
Solution Diagram:	 Data Controller Data Subject Collect data Select default for usage data Allow user to select/filter usage data Provide loggings, reporting and accesses Provide notification on usage data Select other relevant information Aggregate information and present it Allow user to download the information Summaries of collected data Predictive models Visualizations Statistics *Figure 27 - Diagram of Privacy Dashboard*
Source:	https://privacypatterns.org/patterns/Privacy-dashboard

When processing personal data, communication is vital for an increase in trust in the service. Controllers want to present just the right amount of information, so the user understands what is going on but is not bombarded with things they may not want or understand. Users may not realize all the risks that come from sharing their data but do not want to be always aware of what is happening to their data since they may start to overlook essential matters.

The solution requires logging, reporting, and storage and providing information about the data processing that may be relevant to the user. It is especially important to record and show the logs and accesses of data that can change, be deleted by users, be forgotten, or may be processed in unexpected ways.

4.6.5. Request for portability Use Case

Associated Use Case: Request for portability

When the subject's personal data is requested, it must be delivered in "a structured, commonly used and machine-readable format" Art 20. Moreover, the data subject has the right to transmit this information to another controller if consent is given, following Art. 6(1)a or Art. 9(2)a. According to the right of portability, the data subject can request for the data to be transferred directly from one controller to another if this is technically feasible.

The entities present in this use case are:

- Data Subject (Client)
- Data Controller
- Third-party controller

Associated GDPR principles:

- Principles of lawfulness, fairness and transparency
- Data subject's right to data portability.

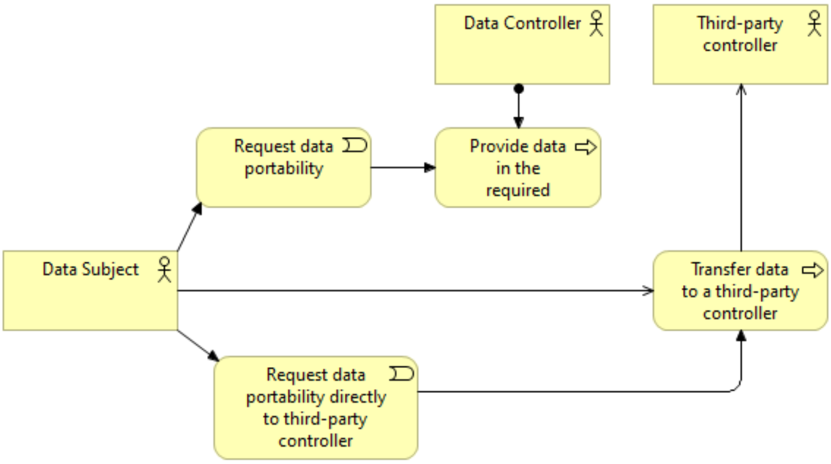
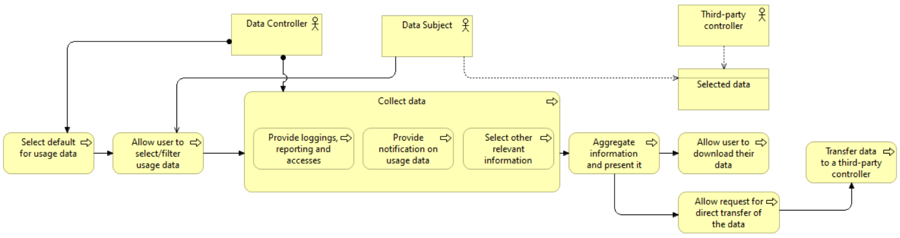

Data Controller
Third-party controller
Request data portability
Provide data in the required
Data Subject
Transfer data to a third-party controller
Request data portability directly to third-party controller

Figure 28 - Diagram of Request for portability use case

Table 21 - Privacy Dashboard (adapted)

Name:	Privacy Dashboard (adapted)
Context:	A service (or product) which processes personal data of users may make that data accessible to them. This is often the case whether conforming to laws about self-determination and notice, or merely wanting to provide an additional privacy consideration for the sake of users. The controller concerned wants to open up about the data they have processed, and to improve the ease of use for configuring privacy settings**.**

Problem:	A system should succinctly and effectively communicate the kind and extent of potentially disparate data that has been processed. Users may not remember or realize what data a particular service or company has collected, and thus can't be confident that a service isn't collecting too much data. Users may want to have that information in their possession; they may also need that data for another service, so a transfer to another controller may be requested. Simultaneously, users may easily be overwhelmed by access to raw data without a good understanding of what that data means, so a simple and easy to read format is important. *Forces and Concerns:* • Controllers want to provide users with sufficient information to determine how it is used, and to prevent regrettable sharing decisions • Controllers want to prevent both over and under-sharing, so as to provide users with the best experience possible • Users often do not realize the privacy risks in providing their personal data or how much data has been processed • Users do not want to be subjected to too many or overly detailed notifications, as they will quickly make a habit of overlooking them • Users may want to see what data has been processed and want to share it with another controller.
Solution:	Provide successive summaries of collected or otherwise processed personal data for a particular user, representing this data in a meaningful way. Provide the ability for the user to receive this information in a structured, commonly used, and machine-readable format (for example, excel sheets or CVS). [Implementation] Implementing this pattern is a matter of providing logging, reporting, and other informational access and notifications on user-selected/filtered appropriately defaulted relevant usage data. It is also essential to give the possibility to download the data. The user can choose to send it to another controller or request for the data to be directly transferred from one controller to another.
Solution Diagram:	 Data Controller Data Subject Third-party controller Selected data Collect data Select default for usage data Allow user to select/filter usage data Provide loggings, reporting and accesses Provide notification on usage data Select other relevant information Aggregate information and present it Allow user to download their data Transfer data to a third-party controller Allow request for direct transfer of the data *Figure 29 - Diagram of Privacy Dashboard (adapted)*

Source:	adapted from https://privacypatterns.org/patterns/Privacy-dashboard

Since this pattern focuses on collecting the data and other information related to the processing of data, it already had an excellent base to use in the use case of portability, only needing a few adjustments.

It has to provide the user the ability to receive this information in a structured, commonly used, and machine-readable format (for example, Excel sheets or CVS). It should also give the possibility to download the data, for the user to send it, or request for the data to be directly transferred from one controller to another.

4.6.6. Request for erasure of data Use Case

Associated Use Case: Request for erasure of data

The data subject has the right to have the data concerning him/her erased if the terms in Art 17 (1) are met without unjustified delays. The controller has to ensure that all the data is deleted while considering the available technology and its implementation cost. They must inform the processors of what was requested to be erased.

The entities present in this use case are:

- Data Subject (Client)
- Data Controller

Associated GDPR principles:

- Purpose limitation
- Principles of lawfulness, fairness, and transparency
- Data subject's right to erasure or "right to be forgotten".

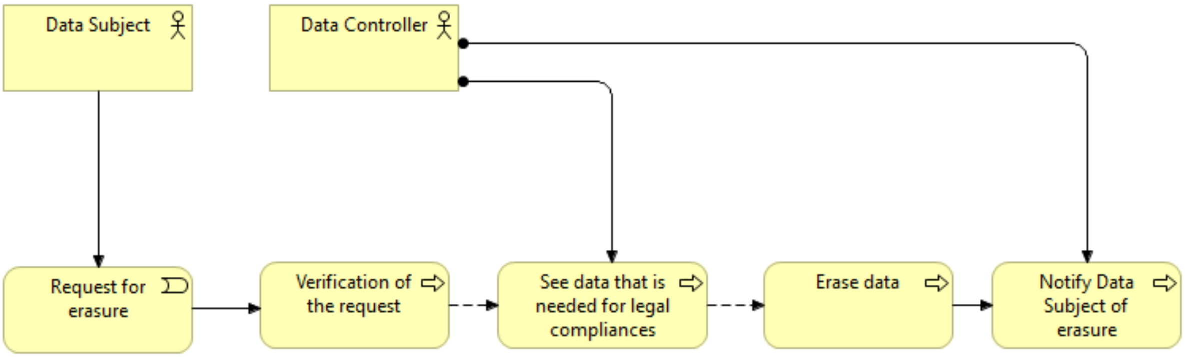
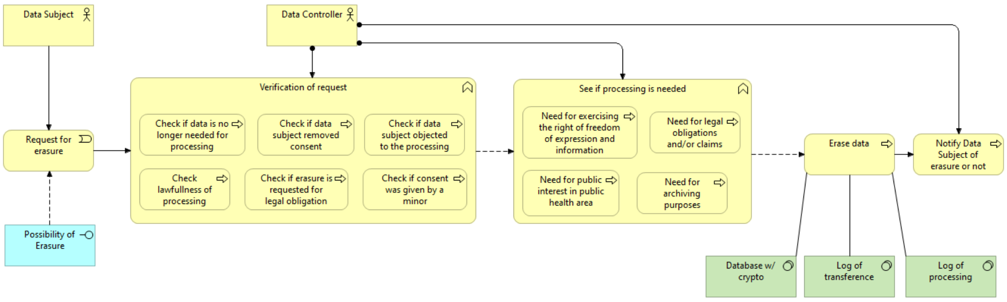

Data Subject
Data Controller
Request for erasure
Verification of the request
See data that is needed for legal compliances
Erase data
Notify Data Subject of erasure

Figure 30 - Diagram of Request for erasure of data use case

Table 22 - Technical Solution for Right of Erasure

Name:	Technical Solution for Right of Erasure
Context:	Services that process personal data want their users to trust in them and comply with the new regulations. In some cases, the users do not want to use the services anymore or may want their data to be removed from the service database for various reasons.
Problem:	Since the appearance of the GDPR, many users are familiar with the right to be forgotten but may not know precisely when to use it. The controller then has to make sure that the request is liable and that if some unlawful processing is occurring, it should be addressed and resolved. It is also vital that the data needed for legal concerns cannot be deleted. The user needs to be made aware of these situations to be transparent (a store cannot delete a purchase registry because of financial constraints). A controller may not want the users to withdraw their data, but it is crucial to guarantee the user that their rights are being fulfilled and increase the trust with the user. *Forces and Concerns:* • Users want to have the possibility to have their data removed not only from processing but also from the services database entirely. • Users may not fully understand in what terms they can request for the erasure of their data. • Controllers want to ensure the users trust in their service. • Controllers need to check if the data has reasonable reasons for its erasure. • Controllers must verify if the data is required for legal claims and if so, they need to notify the user about it.
Solution:	The first step to take is to create an interface that enables personal data's subsequent erasure. Data of individual persons must be retrievable and separately erasable. Subsequent reproduction of the data after deletion is not permitted. After the data subject requests for the erasure of their data, the controller must assure that the request has the right grounds for the erasure to be conducted. If one of the grounds is met, then the controller must check if some of the data needs to be kept to comply with legal obligations. When all of the erasure requirements are met, the data then has to be tracked and deleted from the services database, as well as the logs related to the data subject in question, them a notification of the deletion is sent. When no ground is encountered or other obligations require the data to be kept, the data subject should be notified of the matter.

Solution Diagram:	 Data Subject Data Controller Request for erasure Possibility of Erasure Verification of request Check if data is no longer needed for processing Check if data subject removed consent Check if data subject objected to the processing Check lawfullness of processing Check if erasure is requested for legal obligation Check if consent was given by a minor See if processing is needed Need for exercising the right of freedom of expression and information Need for legal obligations and/or claims Need for public interest in public health area Need for archiving purposes Erase data Notify Data Subject of erasure or not Database w/ crypto Log of transference Log of processing *Figure 31 - Diagram of Technical Solution for Right of Erasure*
Source:	adapted from Privacy Control Patterns for Compliant Application of GDPR [19]

The solution proposed in [19] is about providing an interface that allows the data subject to request their data's erasure. To complete this pattern, we analyzed Article 17 of the GDPR. The "verification of request" function in the diagram (Figure 31) is related to Art.17(1), and the function related to the necessity of the data for processing is based on Art.17(3). The controller must verify the grounds for erasure are valid, so they should be requested, and check if the data processing is necessary.

4.6.7. Request for/and update of data Use Case

Associated Use Case: Request for/and update of data

Personal data must be up to date, so the data subject has to right to request for his/her incomplete or wrong information to be updated (Art 16). There should be a way for the data subject to see and update their information. Suppose the subject changes the email or other relevant personal data; the system should provide an easy way to update the stored/processed data. The logs must also be updated.

The entities present in this use case are:

- Data Subject (Client)
- Data Controller
- [Data Processor]

Associated GDPR principles:

- Trueness and Accuracy

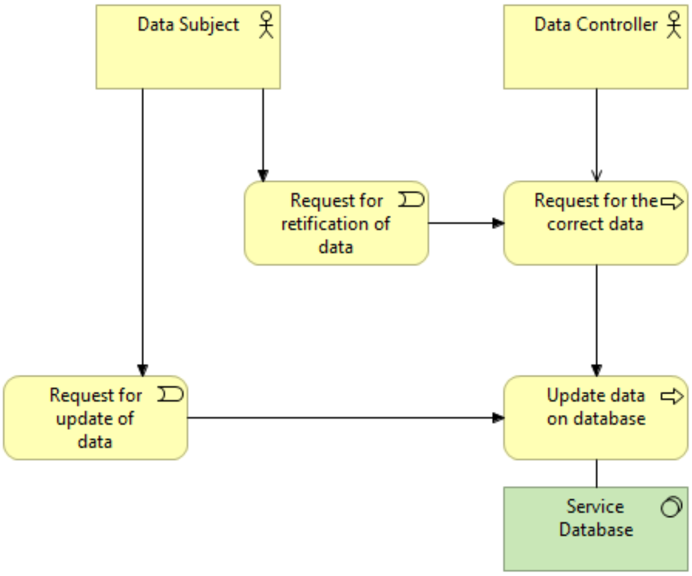
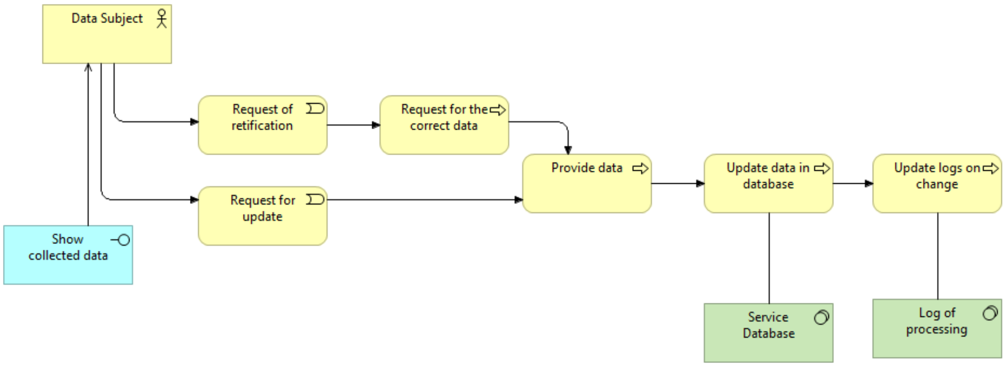

Data Subject
Data Controller
Request for retification of data
Request for the correct data
Request for update of data
Update data on database
Service Database

Figure 32 - Diagram of Request for/and update of data use case

Table 23 - Technical Solution for Update of data

<u>Name:</u>	Technical Solution for Update of data
<u>Context:</u>	When using a service that collects and processes personal data, it is vital to be aware that data can change over time or may be incorrect. To keep the trust of the users who provide their data to the service, incongruities or false information should be rectified for lawful processing of the data.
<u>Problem:</u>	Users may change their email, address, or other personal information and want to update that information to keep it accurate. In some services, the data could have been collected through unlawful channels or incomplete, so it is important to identify and rectify the incorrect data. *Forces and Concerns:* • Users need to be aware of what personal data was collected and is being processed. • Users want their data to be correct. • Controllers want to ensure the users trust in their service. • The data needs to be updated.
<u>Solution:</u>	An interface that shows the user their data with an edit option in the fields is one way to solve this situation. The other suggestion is with an option to request the data to be updated. The user then must provide the correct data or the rest of the data (in case of incomplete data).

	The data should be immediately updated in the service database, and the rectification should be stored in logs kept for the processing of data. If the data is processed by a third-party, they should be made aware of the update.
Solution Diagram:	 Data Subject Request of retification Request for the correct data Provide data Update data in database Update logs on change Request for update Show collected data Service Database Log of processing *Figure 33 - Diagram of Technical Solution for Update of data*
Source:	adapted from Privacy Control Patterns for Compliant Application of GDPR [19]

This pattern's base is similar to the previous one. It has an interface that enables the data subject to see the personal data and later request an update or rectification of inaccurate data. The data then must be updated in the service's database, and the logs must be updated with the instance of the modification.

4.6.8. Change of data processing purpose Use Case

Associated Use Case: Change of data processing purpose

Policies or new purposes for the processing of data may appear, and when it does, a new consent (or update of the previous) must be made. Explicit consent and the right to object are necessary, as well as an update of the records of the processing activity, aka, logs (Art. 30).

The entities present in this use case are:

- Data Subject (Client)
- Data Controller

Associated GDPR principles:

- Purpose limitation
- Integrity and confidentiality
- Principles of lawfulness, fairness, and transparency

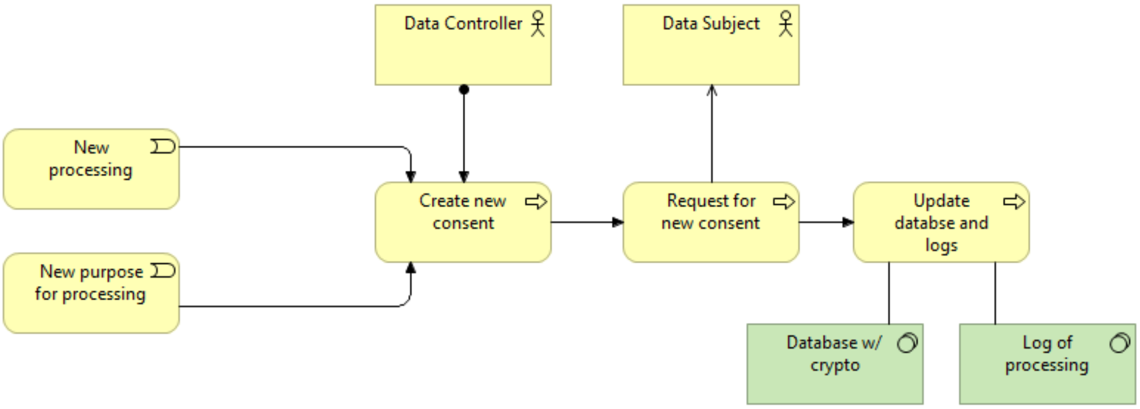


Figure 34 - Diagram of Change of data processing purpose use case

For this use case, the selected pattern is the Negotiation of Privacy Policy already described in Table 17; it was also selected for the Request for restriction on personal data's processing use case.

This pattern was already used for another use case, the request for restriction on personal data's processing, but it also covers this use case's problems.

One of this pattern's propositions is to have opt-in/opt-out choices, allowing users to restrict or grant permissions data processing. Another concern is when new purposes appear, new consent must also be requested, so when this occurs, the user is again given the opt-in/opt-out choices for the new purposes. The permissions are then stored and updated whenever the user changes them.

4.6.9. Consent of minors Use Case

Associated Use Case: Consent of minors

If the data subject is younger than 16 years old (for Portuguese legislation), the consent must be provided by who takes parental responsibility for the child (Art .8).

The entities present in this use case are:

- Data Subject (minor)
- Data Subject's Guardian
- Data Controller

Associated GDPR principles:

- Integrity and confidentiality
- Principles of lawfulness, fairness, and transparency
- The concern about consent

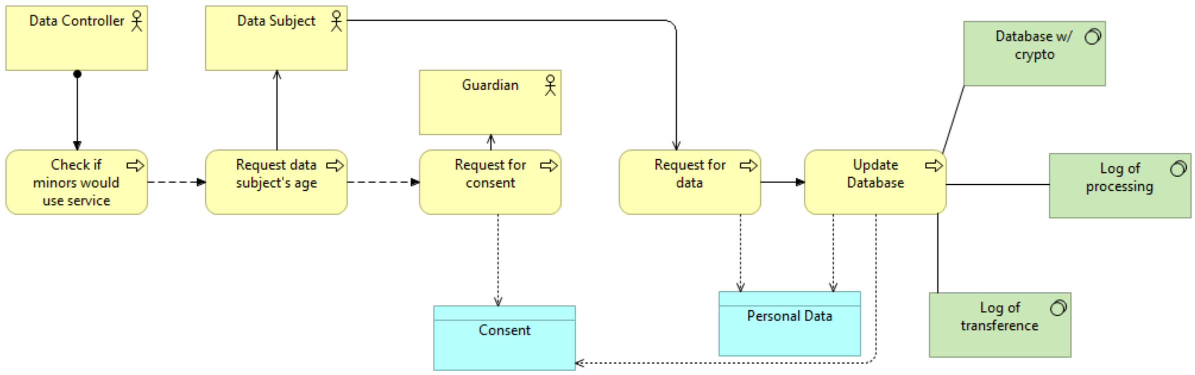
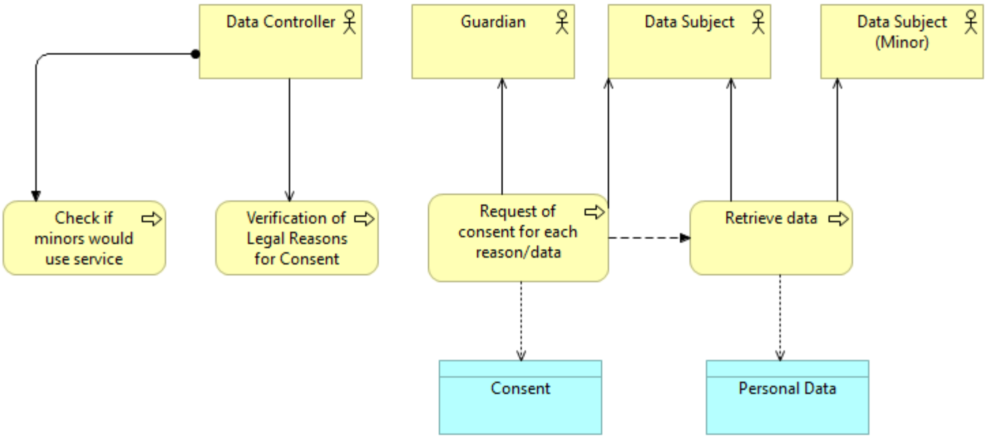


Figure 35 - Diagram of Consent of minors use case

Table 24 - Lawful Consent (adapted)

Name:	Lawful Consent (adapted)
Context:	Where data controllers (e.g. organizations) aim to provide a service (or product) to users, there may be opportunities to reuse data, gather feedback, or make use of user data to further their system's value. Many controllers seek to continually collect and utilize this data, often in ways which warrant privacy concerns. For any data processing (including collection), controllers should first obtain consent from the users in question. There are social norms surrounding the use of personal data which need to be adhered to if a controller wishes to avoid scrutiny. Users do not inherently trust controllers to handle their personal data with care for privacy. Without clearly defined boundaries, these users may have justifiable concerns about what is learned about them, and how this information may be used. Additionally, various jurisdictions supply varying compliance requirements, and these controllers need to cater to every market they provide to. Doing otherwise, possibly by disinterest or negligence, may have financial consequences in addition to potential public outcry. Despite this, controllers regularly consider the impact that their decisions may have on competitive edge and resulting profits. The link between better decision making, possibly less sharing, and reduced monetary gains sways some controllers into unlawful forms of consent. (…)
Problem:	A controller aims to maximize the value of their services by gathering as much sharing and participation as possible, potentially seeing user consent as a barrier to functionality and efficiency. They may inadvertently subvert notions of consent by unnecessarily bundling together desirable services with needs for personal information, or downplaying the significance of the data involved. They undermine self-determination at the risk of losing trust from their users, and attracting legal investigations which may rule their practices unlawful.

	Forces/Concerns: • Controllers want to encourage participation, and thus may be less concerned with investigating or revealing tradeoffs • Controllers may be tempted to bundle various services under a single broad consent request, pressuring users into agreements they might not otherwise accept • Users often want to make use of new and exciting features, and therefore easily overlook downplayed privacy risks • Some users avoid certain services as they realize the potential privacy risks are not being acknowledged • Users may be children
Solution:	A user should be given every opportunity to assess their sharing choices prior to making their consent. The controller should aid the user in comprehending the tradeoffs apparent in using each of their services, without over-burdening the user. These consented services should be purposed-separated, so that users may make use of functionality without first granting unnecessary consent. Rationale: Controllers need to ensure that anything they do with a user's sensitive or potentially identifying data is legal. This pertains to lawfully obtained consent, for purposes which are clear and lawful in their own right. Additionally, anything they do should be resistant to backlash from users. [Implementation] Separate Purposes: Services should be separated into distinct processes for which distinct consent is acquired. Each purpose requires its own consent. These permissions need to be given subsequent to ascertaining sufficient awareness in the user about the consequences of that consent. Freely Given Consent: The users should not be pressured into providing consent. Instead, the benefits may be presented along with the trade-offs so that the users may make an informed decision. In some services, not all users are necessarily capable of making these decisions themselves (e.g. children) and thus provisions need to be made to cater to this, as obtaining the consent from their guardian (whom takes parental responsibility over the child). The provided information should not be misleading, as coerced consent is not a valid form of permission. One way to present policies in an accessible manner is through comparative examples (e.g. in addition to further detail, what is unique about our privacy policy?). (…)

Solution Diagram:	 Data Controller Guardian Data Subject Data Subject (Minor) Check if minors would use service Verification of Legal Reasons for Consent Request of consent for each reason/data Retrieve data Consent Personal Data *Figure 36 - Diagram of Lawful Consent (adapted)*
Source:	adapted from https://privacypatterns.org/patterns/Lawful-Consent

In this case, an adaptation of another pattern was performed. Lawful Consent is a complete pattern that addresses many concerns related to giving consent freely, well informed, and lawful overall.

It already addresses the case of child consent since they cannot make decisions themselves, so only the parts of the pattern that address and are relevant to the minors' use case were selected. The diagram also focuses on the case of the user being a minor and the processes that have to happen in order for the consent to be lawful.

4.6.10. Transfer data processing to a third-party Use Case

Associated Use Case: Transfer data processing to a third-party

In some cases (like subcontracts), the processor or controller share the data with a third-party. When this happens, the data subject must be notified of such actions. For this transfer to be valid, the Commission has to decide that the third-party is trustworthy, i.e., follows the processing policies/restrictions present in the GDPR legislation. In the absence of a decision by the Commission, the transfer can also happen if “the controller or processor has provided appropriate safeguards” (Art. 46) or the data subject has consented to the processing being aware of all the risks (Art. 49). According to article 19, “any rectification or erasure of personal data or restriction of processing” must be notified to whom the data was transferred. It is also important to point out that this is referent to other companies in the same country and a third country or an international organization (Chapter 5, Art. 44-50).

The entities present in this use case are:

- Data Controller
- Third-party

Associated GDPR principles:

- We come across all of the GDPR principles (since all the regulation must be followed)
- Accountability

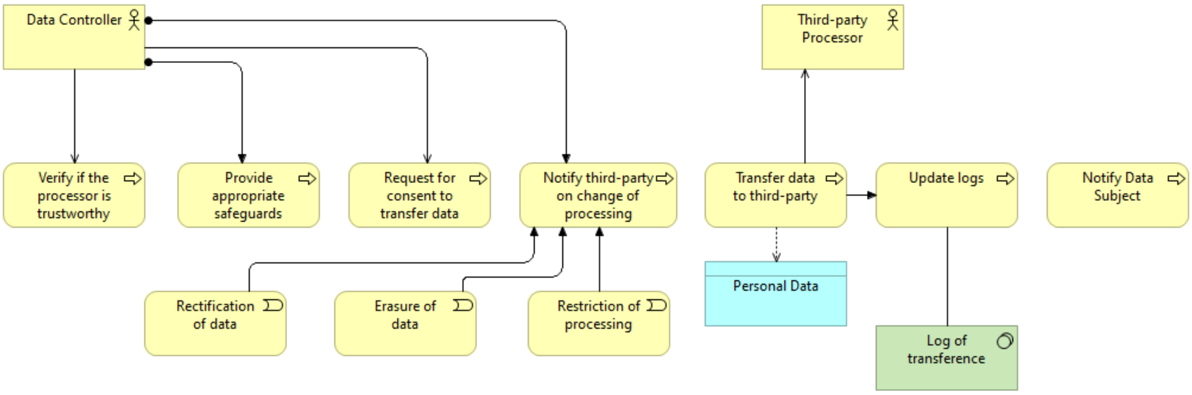
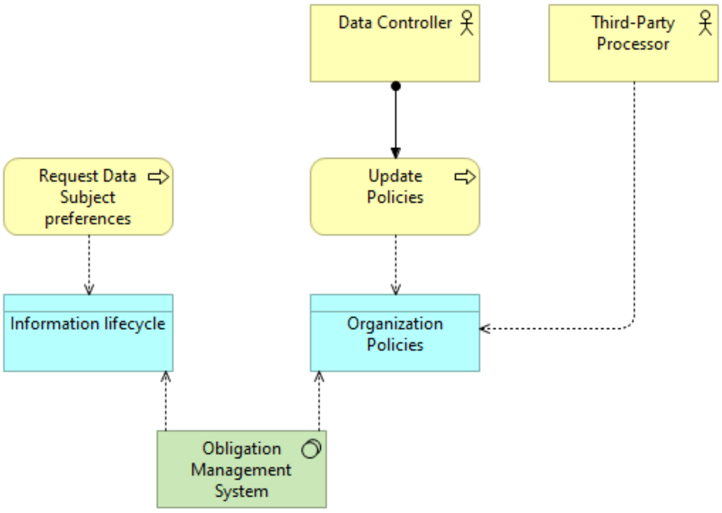

Data Controller
Third-party Processor
Verify if the processor is trustworthy
Provide appropriate safeguards
Request for consent to transfer data
Notify third-party on change of processing
Transfer data to third-party
Update logs
Notify Data Subject
Personal Data
Rectification of data
Erasure of data
Restriction of processing
Log of transference

Figure 37 - Diagram of Transfer data processing to a third-party use case

Table 25 - Sticky Policies or Obligation Management

Name:	Sticky Policies or Obligation Management
Context:	Multiple parties are aware of and act according to a certain policy when privacy-sensitive data is passed along the multiple successive parties storing, processing and sharing that data.
Problem:	Data may be accessed or handled by multiple parties that share data with an organization in ways that may not be approved by the data subject
Solution:	Service providers use an obligation management system. Obligation management handles information lifecycle management based on individual preferences and organisational policies. The obligation management system manipulates data over time, ensuring data minimization, deletion and notifications to data subjects. The goal of the pattern is to enable users to allow users to control access to their personal information. Examples of policy specification languages include EPAL, OASIS XACML and W3C P3P. Tracing of services can use Identifier-Based Encryption and trusted technologies

Solution Diagram:	 Data Controller Third-Party Processor Request Data Subject preferences Update Policies Information lifecycle Organization Policies Obligation Management System *Figure 38 - Diagram of Sticky Policies or Obligation Management*
Source:	https://privacypatterns.org/patterns/Sticky-policy https://privacypatterns.org/patterns/Obligation-management

This pattern has two names on the website, but it is the same. The concern is that in many cases, the processing or part of it is performed by third parties. The users may be aware of the processing by the organization that provided the service, but not on the other parties, and may disapprove of such.

The idea is to have a system that manages data based on the user's preferences and based on organization's policies. These measures ensure that only the needed data is processed and that the user's rights are guaranteed by allowing them to control who can access their personal data. The third parties also must follow the guidelines the company creates.

Table 26 - Trust Evaluation of Services Sides

Name:	Trust Evaluation of Services Sides
Context:	When using a service (or product) offered by a controller, the level of trust held by users is crucial. Without sufficient trust, the users would seek alternatives or generate bad publicity. They will use a system more cautiously, regardless of whether it is necessary. In many systems this lessens the quality of service offered, not only to the user in question, but holistically.
Problem:	Users want to have reason to trust that a service does not undermine their personal privacy requirements. They do not want to have to take controllers, and third parties, at their word alone. *Forces and Concerns:*

- Controllers, as well as third parties, want to show that they are provably trustworthy and reliable
- Less confident entities will not make this effort alone
- Users want to verify claims which controllers and third parties make without having to do so themselves
- Users benefit from a standardised way of indicating trust, as it is easier and quicker to look into if done consistently and often.

Solution:

Supply a function which informs users of the trustworthiness and reliability of services, and that of the third parties connected to those services. These qualities may be determined, and assured, through independent evaluation of given criteria.

[Structure]

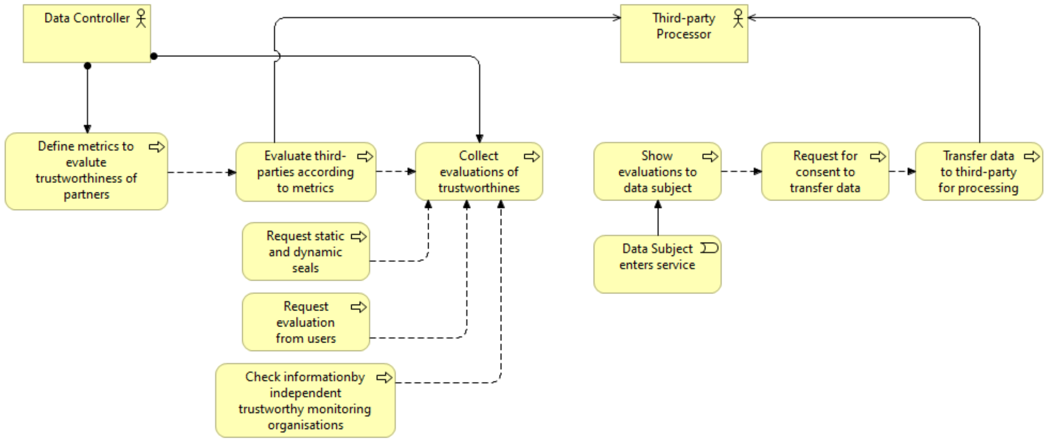
Information regarding a service's trustworthiness and reliability needs to be clearly indicated to the user prior to or during collection. It may therefore be brought up along with obtaining informed consent. This ensures that the user does not make misinformed or uninformed decisions, especially as this can seriously jeopardise trust. A visual highlight which succinctly asserts this quality may also be displayed in persistent manner, or where otherwise contextually relevant.

[Implementation]

A trust evaluation function should be based on suitable parameters for measuring the trustworthiness of communication partners and for establishing reliable trust.

[Trust] in a service provider can be established by monitoring and enforcing institutions, such as data protection commissioners, consumer organisations and certification bodies. Privacy seals certified by data protection commissioners or independent certifiers (e.g., the EuroPrise seal, the TRUSTe seal or the ULD Gütesiegel) therefore provide especially suitable information for establishing user trust. Such static seals can be complemented by dynamic seals conveying assurance information about the current security state of the system and its implemented privacy and security (PrimeLife) functions. Further information sources by independent trustworthy monitoring organisations that can measure the trustworthiness of services sides can be blacklists maintained by consumer organisations or privacy alert lists provided by data protection commissioners.

[Also,] reputation metrics based on other users' [ratings] can influence user trust. Reputation systems, [for instance] in eBay, can however often be manipulated by reputation forging or poisoning. Besides, the calculated reputation values are often based on subjective ratings by non-experts, [through which privacy-friendliness may be difficult to judge].

	A trust evaluation function should in particular follow the following design principles: – Use a multi-layered structure for displaying evaluation results. – Make clear who is evaluated. – Inform the user without unnecessary warnings. – Use a selection of meaningful overall evaluation results.
Solution Diagram:	 Data Controller Third-party Processor Define metrics to evalute trustworthiness of partners Evaluate third-parties according to metrics Collect evaluations of trustworthines Show evaluations to data subject Request for consent to transfer data Transfer data to third-party for processing Request static and dynamic seals Request evaluation from users Check informationby independent trustworthy monitoring organisations Data Subject enters service *Figure 39 - Diagram of Trust Evaluation of Services Sides*
Source:	https://privacypatterns.org/patterns/Trust-Evaluation-of-Services-Sides

When processing data, especially personal data, it is essential to pass to the user that they can trust in the service. This proof of reliability should be given to the user without searching for it themselves and needs to come from all the parties that process data.

The solution comes from informing the users of the service's trustworthiness and reliability and all of the parties that work with it. When data is being collected, this information needs to be shown to the user; it could be when requesting consent. This information is created by creating measurements that evaluate the parties' work. Ratings are created based on evaluations of those measurements, and the summary of the scores or performance is presented to the user. One of the metrics could be user ratings.

4.6.11. Notify the data subject Use Case

Associated Use Case: Notify the data subject

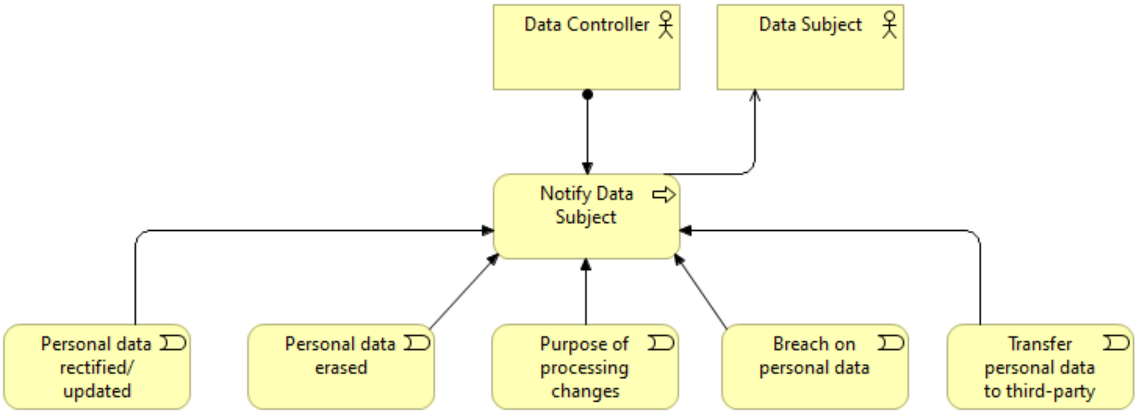
After each change in the personal data, the data subject should be notified. If the purpose for processing the data changes or a new reason appears, the data subject should be notified of such occurrences. The data subject's notification also happens if the data is updated (by request or not of the data subject) or is erased (by the termination of the storage term, request of the data subject, or unsubscribing of services, for example). Moreover, any breach or transfer to another processor must be informed to the data subject.

The entities present in this use case are:

- Data Subject (Client)
- Data Processor
- [Data Controller]

Associated GDPR principles:

- Principles of lawfulness, fairness, and transparency



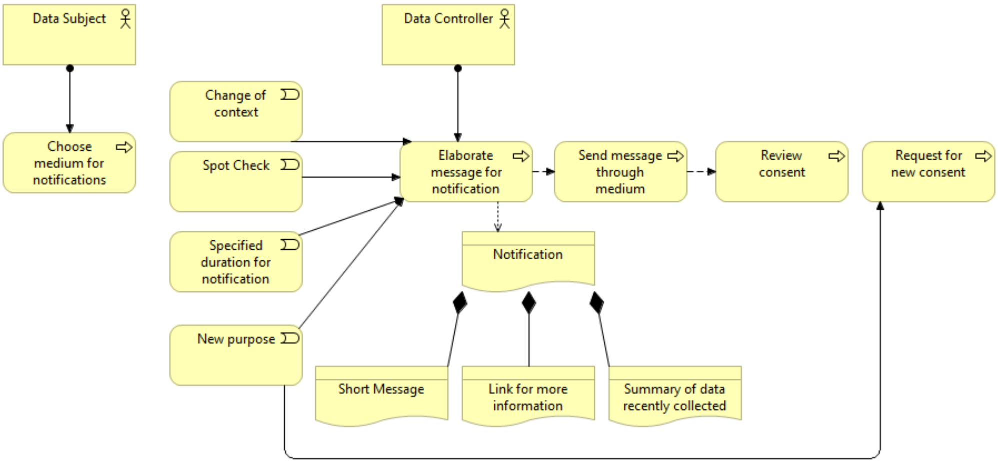
Data Controller
Data Subject
Notify Data Subject
Personal data rectified/ updated
Personal data erased
Purpose of processing changes
Breach on personal data
Transfer personal data to third-party

Figure 40 - Diagram of Notify the data subject use case

Table 27 - Asynchronous notice

Name:	Asynchronous notice
Context:	Many sensor related or other recurring forms of data collection are important for improving service (or product) quality, but occur in a manner which is not apparent to the user. Even where the user is informed of such processing, the nature of that processing may cause it to occur within contexts the user would not consent to. Users are also subject to forgetfulness. The controller processing this information therefore seeks to ensure that consent is retained. Some interfaces necessitate more restrictive use of screen real estate, however, and as such can not accommodate extensive information or persistent elements.
Problem:	Users being tracked and monitored may not consent to processing they had previously consented to, as the context surrounding that processing is subject to change. Also, initial consent may have been forged by an attacker or have been provided by another user of a shared device -- if synchronous notice is only provided at the time of consent, a user may inadvertently distribute personal information over a long period of time after having lost control of their device only momentarily.

	Forces and Concerns: • Users may change their minds or forget about consent they have given • Users may not realize the processing they consented to is currently in effect, potentially allowing collection of information they would not want collected • Controllers do not want to collect data for which consent is uncertain, where users may feel violated or otherwise let down • Controllers cannot remind users of their consent all the time Providing an asynchronous notice requires a reliable mechanism to contact the user (a verified email address or telephone number, for example). Care should be taken to ensure that the mechanism can actually reach the person using the device being tracked. (For example, notifying the owner of the billing credit card may not help the spouse whose location is being surreptitiously tracked.) In contrast to the common privacy practice of providing consistent and reliable systems, you may wish to provide random asynchronous notice. If there is a concern that a malicious user may have opted-in the user without their knowledge, a notice that is sent once a week at the same time each week may allow the attacker to borrow the device at the appointed time and clear the notice. Many repeated notices may annoy users and eventually inure them to the practice altogether. Take measures to avoid unnecessary notices and some level of configuration for frequency of notices. This must be balanced against the concerns of an attacker's opting the user in without their knowledge
Solution:	Whenever there is a context switch, sufficient duration, or random spot check, provide users with a simple reminder that they have consented to specific processing. The triggers and means for contacting the user may be chosen by the user themselves, who should be able to review and if necessary retract their consent. [Implementation] Implementation depends on the medium chosen for conveying the notification, and also on the service facilitating collection. For mediums with less space, shorter messages should be provided, but even in more traditionally long-winded options such as email, brevity should be favored. The user should be able to obtain more information by a linking mechanism, also dependent on the medium. The most important information to provide is the fact that they have consented to specific data for specified purposes, and that a context change, spot check, or specified duration has triggered the reminder. Context changes are most notable, as these are most likely to affect the consent. Note that changes to purposes and means instead require new consent, not merely notification.

	Asynchronous notices may also include a summary of the data recently collected (since the last notice, say) in order to provide clarity (and reminders) to the user about the extent of collection. By ensuring that users aren't surprised, asynchronous notice may increase trust in the service and comfort with continued disclosure of information
Solution Diagram:	 Data Subject Data Controller Change of context Choose medium for notifications Spot Check Elaborate message for notification Send message through medium Review consent Request for new consent Specified duration for notification Notification New purpose Short Message Link for more information Summary of data recently collected *Figure 41 - Diagram of Asynchronous notice*
Source:	https://privacypatterns.org/patterns/Asynchronous-notice

Users may consent to the processing but can forget they did so, or the consent may have been given falsely. If the notice is given only when consent is requested, this does not give a very reliable perception of the service. Besides, if something changes or there is suspicious activity, this should be informed to the user.

First, a reliable medium for the user to be notified must be provided (for example, a second email). The message's structure must depend on the medium, and additional or more detailed information could be given through a link to another page. Whenever there is a change in purpose or context of processing previously agreed with the data subject, they should be notified of such (with a request of new consent if pertinent). This pattern can also be used to inform users of other things, like deletion or update of the data.

Another selected pattern is the Unusual Activities, which is described in Table 16. We already included this pattern in the Inform of Breach use case, but it is also very relevant to this one. One of the main things the data subject must be notified of is if their data is at risk. This pattern provides an excellent logic to this matter that could be applied to other cases of notification. The tracking of the logins or changes in the environment could be used for tracking changes in purposes or transfers, updates, etc., of the data. Some of the different ways the user can access and assure their identity can also be used as notification channels.

5. Demonstration

The primary goal is to help companies understand how the GDPR affects the design of their services and what are some of the solutions for it.

For demonstration purposes, we will use the case of a platform where the patterns could have used. Consider a platform that works as a channel for other organizations' platforms through a login and has an access log history.

5.1. Description of Platform

This platform can be divided into various use cases or functionalities. The first is the sign up of the user on the platform. When the user first logins into the platform, the email or other identification data are requested, as well as the password. The data subject can choose to add the name to the profile later, but it is not required. It must also give the option to update the email or the other identifier. The user can later choose to delete their profile if they choose so by selecting a "Delete account" button.

Another platform's functionality is to provide access to other platforms by linking to the home page. The user has the list of the sites they can access and choose the one they want to enter. The platform also registers every access and provides a history log of it. The information displayed is the site they accessed, the time (if it was that day), or the date. If the account has a suspicious login, it must handle it.

We will call these use cases Login of the user, Add the name, Update the profile, Delete profile, Access to other platforms, Suspicious login, and Access history logs.

5.2. Selecting Use Cases

Let us start by going through the library and the use cases and see if they are relevant to the service; if not, there is no point in seeing the patterns for those use cases, and the search becomes faster.

The first use case is very relevant since the access to the platform is through login. Since breaches can occur, the use case related to them may also be relevant. Suppose the only processing that is being performed is the login and the history of the accesses. In that case, the restriction of processing may not be applicable, as well as the request of portability, since it makes no sense to provide the login credentials of one service to another. The request for erasure and access to the data can be significant, plus the update of data (the username or the media through each the authentication is made, can change).

It is unlikely that the platform will do more than what was set, but the architects should think of a plan in case the purpose or the processing changes. Minors will not use this service, so the use case for their consent can be discarded. The notification use case can be relevant, but if it only concerns breaches, this is already covered in the other use case. The platform may only be a door for other services, but a trust bond must be present for a trustworthy relationship between the services, so a look through the use case of transfer processing to a third-party is important.

In short, the use cases we will take into account are: Register in system, Inform of Breach, Request of personal data, Request for erasure of data, Request for/and update of data and Transfer data processing to a third-party.

We can relate the platform and the library use cases like:

Table 28 - Relation between platform and GDPR library use cases

Sign up of the user	Register in system
Add the name	
Update the profile	Request for/and update of data
Delete profile	Request for erasure of data
Access to other platforms	Transfer data processing to a third-party
Suspicious login	Inform of Breach
Access history logs	Request of personal data

5.3. Selecting the Patterns

Now that we established the use cases, we start looking for the patterns that make more sense to apply or implement some parts. Suppose the project is new is more comfortable to implement the pattern to its fullest. If it is a pre-existing work that must be adapted to comply with the GDPR, only some parts can be implemented, but that is one of the good things about patterns; they are not a strict set of rules but flexible solutions.

Considering the first use case, we recognize that the only way the processing can be lawful, in this case, is by consent. The platform may not process that much data, so long and exhaustive explanations and requests may tire the user. However, consent is still necessary, and for it to be lawful, the data subject must be aware of the risks and the purposes for processing. The pattern that makes the most sense and provides a more direct and quick experience for the user is Informed Implicit Consent. The organization must evaluate what data is necessary for the service and write clear and concise explanations. When the data subject registers in the platform, the data is demanded, and the consent is requested implicitly, as cookies settings or a simple check box, for example. An explanation accompanies consent and the option to object to it, in which the impossibility to use the service may be presented.

In terms of purpose limitation and data minimization, the user must trust the service will only collect and process the necessary data. Another concern is that the user understands the purposes so that the difference in information between the data subject and the controller is little. A suitable pattern to follow is Minimal Information Asymmetry. It focuses on the fact that only the essential data should be collected. The reasons for the processing must be written in a clear and easy to understand way so that the level of information on the matter is not very asymmetric between the data subject and the data controller. The pattern selected previously already requires an evaluation and selection of necessary data, making this already accomplished in a way. The users may be given the option to give more data (if it is relevant), as the name, but the base is to keep the data to a minimum. In terms of policies, for each purpose, consent must be written. Since the data requested will be minimal, they became more straightforward and fewer. This makes the user less pressured to read much information and is more likely to read the policies to the fullest and become aware of the risks and the purposes.

For integrity and confidentiality purposes, the controller wants the platform to be secure and reliable. In the library, we have four patterns to choose from, but it must be considered if the work of implementing them is worth it in terms of the data that will be stored. The platform only works as a door to other services that probably have more data stored, so an adaptation of the Encryption with user-managed keys pattern could be made. This adaptation could allow the data subject to choose the encryption keys with some guidelines to create safe keys. The storage of those keys can be in the user's device or, to simplify, in the service provider's database.

In the case of suspicious login, we have the Inform of Breach use case. The problem's base is to identify if the login is suspicious or not; another concern is how to inform the user or provide a way to confirm or deny the identity of the data subject. The Unusual Activities pattern is an excellent choice to solve these questions since it addresses these concerns. Moreover, since other services already use it, it may be more apparent to implement than the other. In this case, the main concern is when a person tries to access the user's account, for whatever purpose it may be, and not so much about data illegally collected from the service provider's database. This pattern is the one that addresses that case the best.

The platform must collect some identifiers and define norms for which accesses to the account may be suspicious, like a different device, browser, country, or operating system. When a login is unusual, the data subject is notified of such and requested to verify the activity by selecting an "it was me" message. It can also be requested another security question, like a second password already predefined or a one-time password sent to the email, for example.

In the case of a request for personal data, not much data needs to be collected and what may be more relevant for the platform is providing logs of access to the platform and other services. This service requires a solution for what information needs to be stored and how to present it so that the user is not overwhelmed. We need a pattern that focuses more on logging; for this, the Personal Data Table is the right pattern. The solution suggests how the logs can be structured, and an interactive table is a way to show these logs and the ones the platform is supposed to show.

For every update, access, and request, the log system must be updated with this information. The structure present in the pattern is to have the type of data, the data itself, the date of when the collection occurred, and who accessed the data. We can add another record with the login, the login date, and the accessed services in that login period for this platform. An interactive table is then presented, allowing the user to see the history, and a button that allows the records to be downloaded should be added.

Regarding the deletion of the profile use case, before the GDPR, the user only had to select an option to delete the account and no concern over if the data was still stored or not existed. Now the accounts can still be deleted, and some information can be kept, but this needs to be informed to the user, as well as the reasons for it. It is easy to select the pattern since there is only one. The technical solution proposes an interface that allows the user to select an erasure option, expressing the request's grounds. These grounds are then analyzed to see if the erasure is valid. If so, the data is deleted (being aware of possible constraints), and if relevant, third parties can be notified of this action. In the case of erasure, the data subject is notified when the process is complete. When the erasure is not possible or acceptable, the data subject is also notified, accompanied by an explanation for the case.

As mentioned above, although the platform is not supposed to collect that much information from the data subject, for login purposes, an email or another contact may be retrieved. With time, an alteration of them may occur. Also, it must be given the possibility to add a name to the profile. To solve the problem of this use case, we also have a Technical Solution. This solution also requires an interface that allows the data subject to edit their data (like their profile, for example). After the data subject changes or completes it, this data should be updated in the service provider database. The occurrence of the update must also be recorded in the logs.

Lastly, since the platform interacts with other parties but does not require them to process data they collected, it is more relevant to select a pattern whose trust concerns are simple and easy to implement. The pattern that addresses this matter better is the Sticky Policies or Obligation Management. This pattern focuses on a system that handles the information life cycle management based on the data subject preferences and organizational policies. The system may be too complex for what the platform needs; however, the organizational policies are an excellent way to provide the third-party with the rules it must follow in order for the partnership to continue trustworthy.

The Trust Evaluation of Services Sides also has a big focus on the trust relationship that has to exist between the third-party and the service provider, but it involves many evaluations. These evaluations are performed not only by the organization but also by the users. Regular checks on the third-party trustworthiness are also required, making this too complex to implement and probably a hazard to the user since the platform is only a portal to third-party services and not a delegation of data processing.

Below is a table that summarizes the patterns selected for each use case.

Table 29 - Use Cases and Patterns Selected

Use Case	**Pattern**
Register in system	Informed Implicit Consent
	Minimal Information Asymmetry
	Encryption with user-managed keys
Inform of Breach	Unusual Activities
Request of personal data	Personal Data Table
Request for erasure of data	Technical Solution for Right for Erasure
Request for/and update of data	Technical Solution for Update of Data
Transfer data processing to a third-party	Sticky Policies or Obligation Management

5.4. Changes on the platform

With the implementation of the patterns, some changes in the processes and some architecture requirements must be made to comply with the GDPR. In this chapter, we will present some of these changes or add-ons, and lastly, a Diagram of the new architecture will be shown.

Let us start by considering the first two patterns, Informed Implicit Consent and Minimal Information Asymmetry. The processes that need to exist are assessing the data that will be used and making sure to request the minimum needed. With clear language, write the purposes for collecting the data and the risks that may come from sharing this data. This data may be an email address and a password; if the data subject wants to give more, as a second email for security reasons or the name for the profile, this should be possible.

Regarding the pattern Encryption with user-managed keys, the controller must write some simple guidelines to create strong encryption keys without overwhelming the user and then use those keys to encrypt the data subject's data before storing it. The keys can be managed and stored by the user and their device or by the service provider.

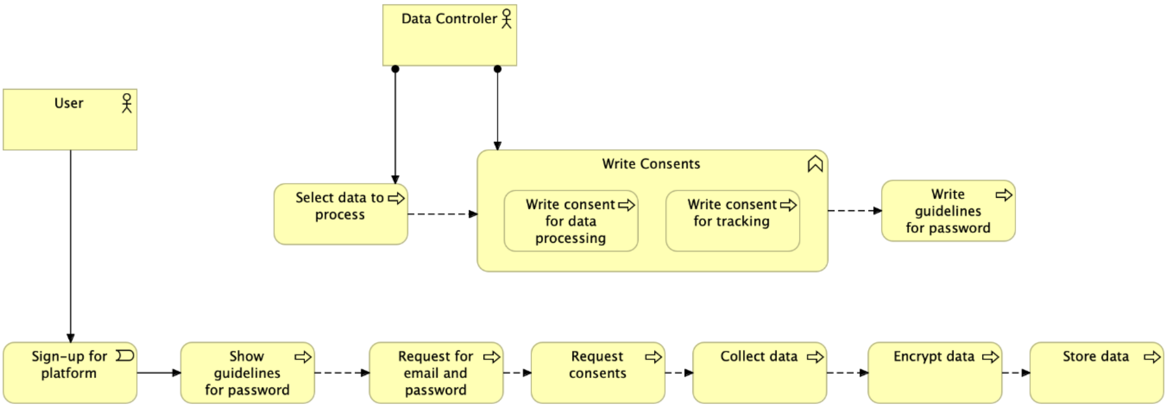

Data Controler
User
Select data to process
Write Consents
Write consent for data processing
Write consent for tracking
Write guidelines for password
Sign-up for platform
Show guidelines for password
Request for email and password
Request consents
Collect data
Encrypt data
Store data

Figure 42 - Diagram for Sign up of the user

The main changes that the patterns brought were the processes related to the consents and the password's guidelines. First, it must be ensured that the requested data will be necessary for the processing and that no further information is collected, and the consents must be requested and stored. The tracking consent is related to the next pattern, but it must be presented when the user signs up to the platform.

For the Unusual Activities use case, access norms need to be formulated. A track of the login must be performed by running native code to collect device information, like the browser, country, region, and device. For this to be made truthfully, consent for such must be requested and stored. When a suspect login occurs, the data subject should be notified via email, for example. For a complete security check, a one-time password can be sent to the data subject to enter the platform, validating the authentication. Both the consents for the collection of data and the tracking could be done by cookies settings.

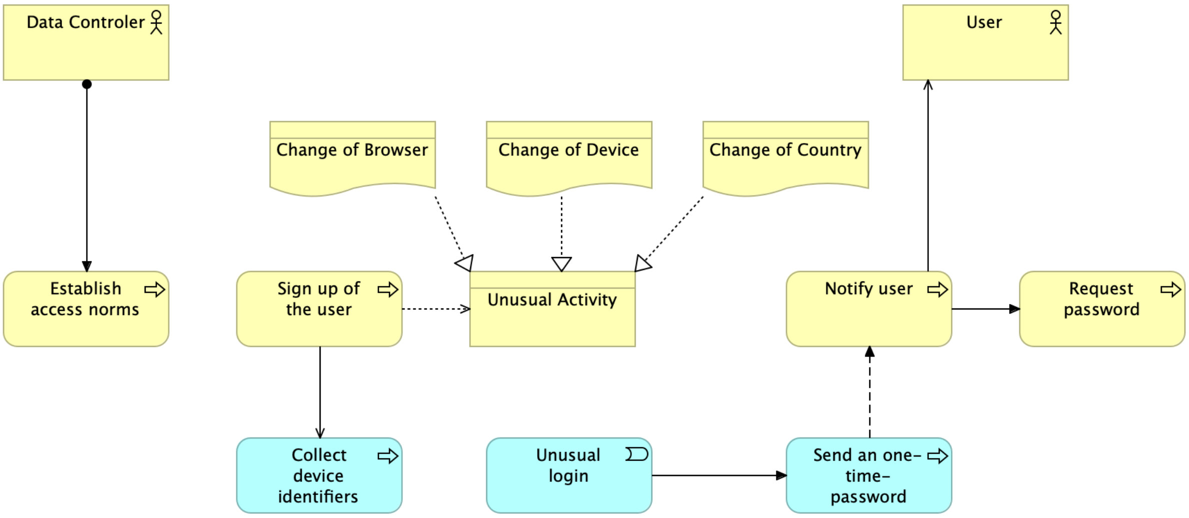


Figure 43 - Diagram for Suspicious login

Many services already use this pattern, so it is probably more familiar to organizations. The main impact was already shown in Figure 42, which is the consent for tracking. After consent, the system collects device identifiers, like the operating system, for example. Whenever a login is performed, it checks for any change on the browser, device, or country. To assure the login is lawful, a one-time-password is sent to the user; this way, if the one who logged in is the user, the user's access to the service can continue.

Since a history of accesses to other services exist, a record system is already present. However, there should also be a record of other aspects like consent storage, when given, and when the data was collected. As was mentioned previously, when choosing the Personal Data Table pattern, a possible structure of the record can be the login, the date record of the login, and the services that were accessed in that login period. Complementing, for every update, request, and possibly any unusual activity, the occurrence of such and the date should be stored. Then, by having an option in the platform, the user can access this information through an interactive table and may even download it in an excel file.

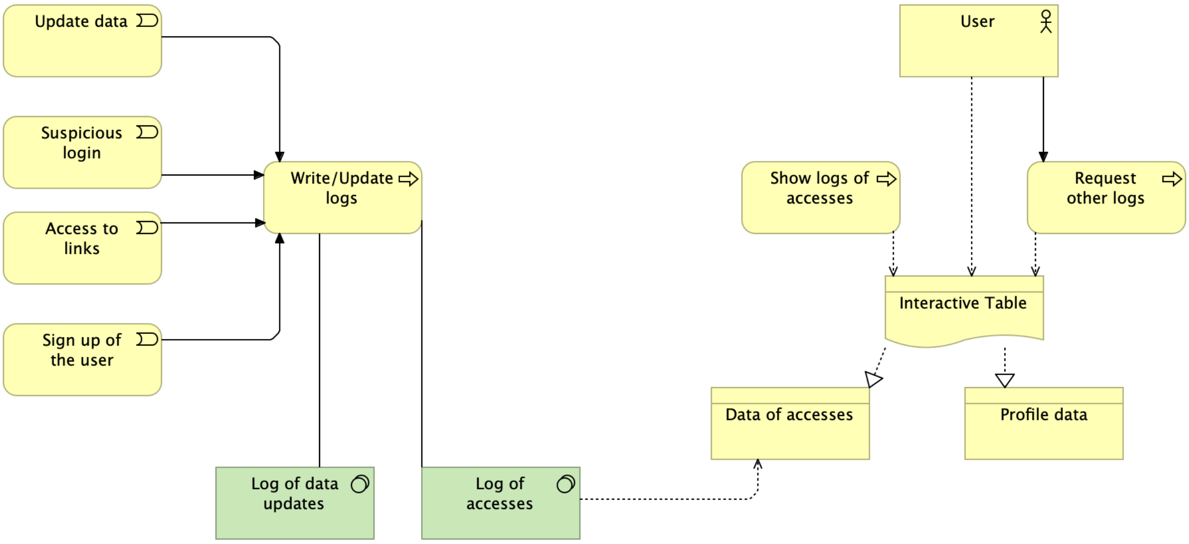


Figure 44 - Diagram for Access history logs

In this case, the logs of access were already something that needed to be implemented since it is a fundamental part of what the service provides. What the pattern adds is the storage of more information related to the user's data. Also, the use of an interactive table as a way to show the information is something that comes from the Personal Data Table. Although not represented in Figure 44, a structure of how the information is stored also comes from the pattern. For the log of data updates, the structure can be the type of data, the data itself, the date of the collection, and who accessed the data. The login, the login date, and the accessed services in that login period can be the structure for the log of access.

For the Technical Solutions patterns, an interface that allows the user to erase their account or other data must be implemented. It should also allow for updates of the data, like a change of the email, password, or to add the name to their profile or change incorrect information. When the data is updated or removed, the logs should record these events, and, if relevant, the third parties can also be notified. When the user requests the erasure of their data, the reason should be evaluated. Whether the request is accepted or not, the data subject must be notified.

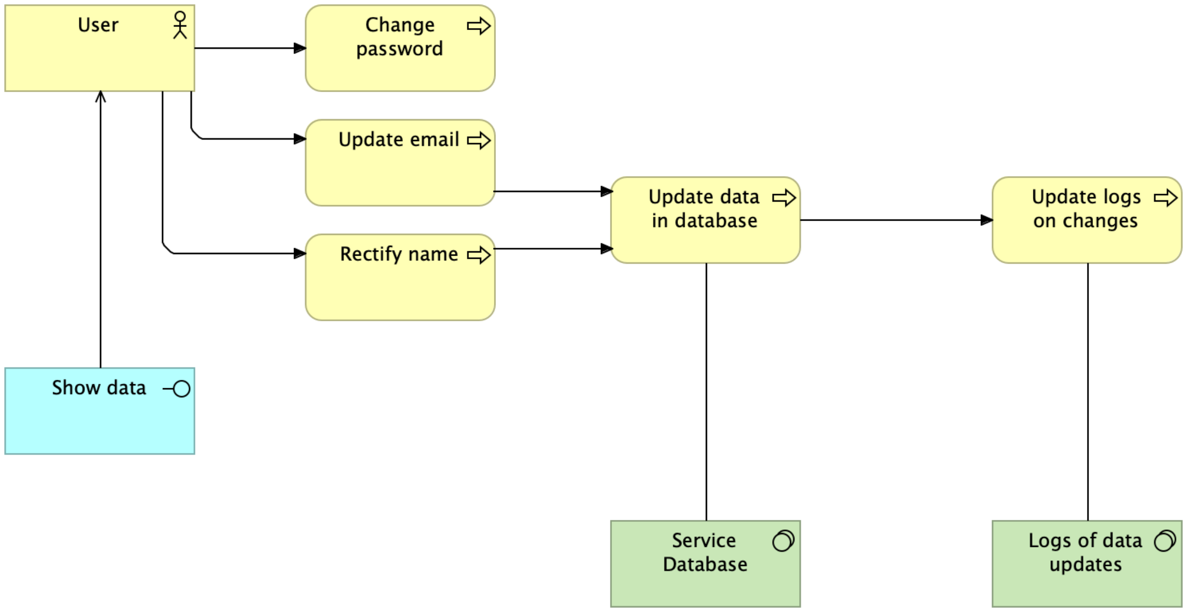


Figure 45 - Diagram for Update the profile

The main change the pattern brings, and that is crucial for compliance with the GDPR, is the process that updates the logs related to the data. Usually, interfaces already allowed users to update their email or change the name, but the updates' occurrences were not stored for lawful processing.

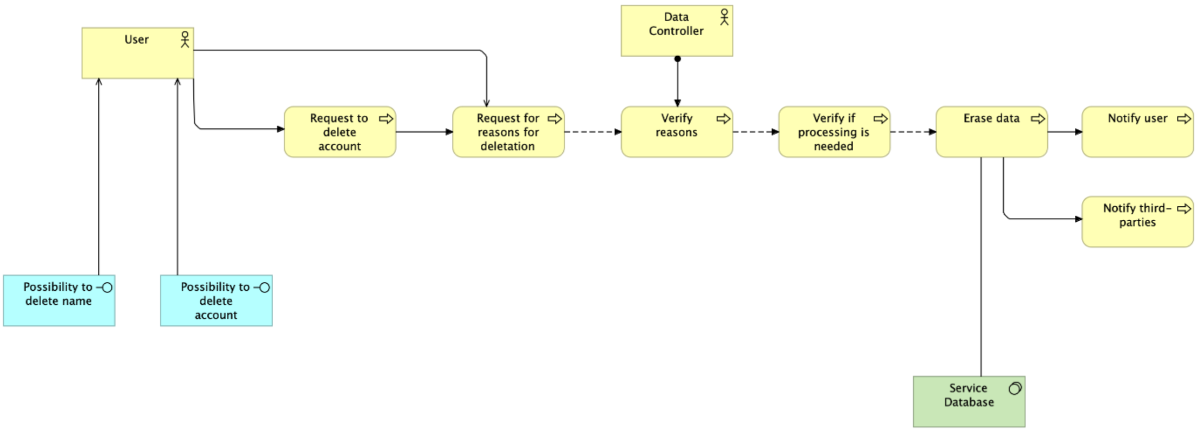

User
Data Controller
Request to delete account
Request for reasons for deletation
Verify reasons
Verify if processing is needed
Erase data
Notify user
Notify third-parties
Possibility to delete name
Possibility to delete account
Service Database

Figure 46 - Diagram for Delete profile

The request for the reasons of erasure and their verification are processes that are fundamental to the regulation. The notification of the user (in case of erasure or not) and the third parties are also crucial for compliance. This diagram simplified the verification processes, but they can be seen to their fullest in Figure 31.

Regarding the Sticky Policies or Obligation Management pattern, the organization policies must be thought and written and then sent to the third parties that interact with the platform. Before collecting the data, the user must be informed of the partnership, possibly when the consent is being requested. Throughout the existence of the relation, the policies must be reviewed and updated as necessary.

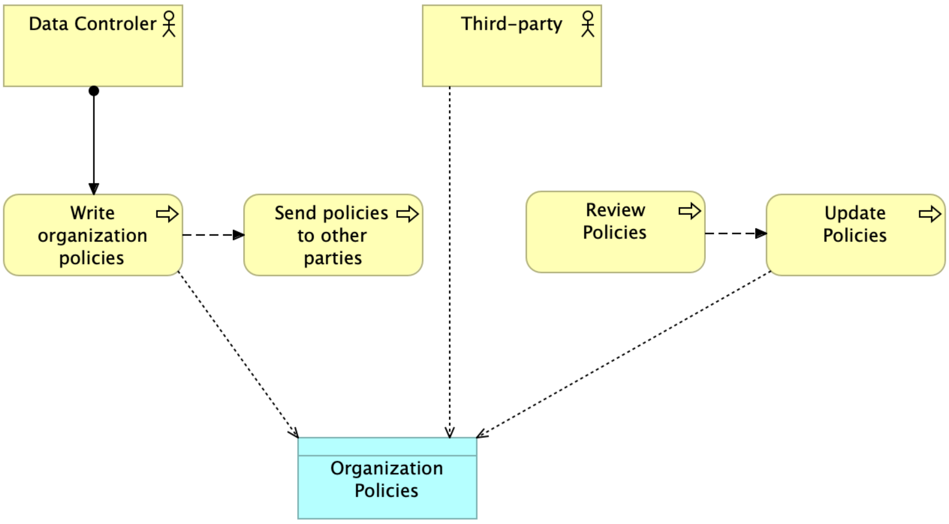


Figure 47 - Access to other platforms

The concern over trust is essential to the GDPR; the data subject must trust that all the processing is lawful and secure. In case processing or sharing of data with a third party, this trust becomes even more relevant. The creation of policies allows organizations to share what they require from others and helps the third parties comprehend what is expected from them. With time, situations may change, so the review and update of the policies are crucial for them not becoming irrelevant.

As we can see, the patterns can be adapted to the services' context and needs while still maintaining compliance with the GDPR. In this case, not all the use cases were relevant to be explored, narrowing the search. If more than one pattern for the use case exists, we can select the one we see more fit and that best addresses the service's concerns in question.

6. Evaluation

This work gives possible solutions for GDPR compliance-related problems. The idea is to provide solutions and give awareness of the constraints and requirements the GDPR brought. Although it is already a well-known regulation, it is hard to comprehend what needs to be executed and what are the things organizations need to look out for when building or updating their services. Hopefully, this work can also help in understanding better what needs to be done in order for their work to be GDPR compliant.

6.1. Differences from other works

In section 3.3, we analyzed some previous work done to assist organizations in achieving GDPR compliance. So what does this work bring new?

Firstly, it is accompanied by a set of guidelines, written after analyzing the steps for GDPR compliance from several sources and combining them with the regulation articles. This way, those with no prior or little knowledge about the subject can already see what needs to change. It can also work as a checklist to ensure that everything was addressed and compliant with the GDPR.

Then, it presents a library organized structure that connects privacy by design patterns with GDPR requirements. It brings possible solutions instead of just one big solution. There is a collection of patterns to choose from, according to the services' characteristics and restrictions. The use cases help check what is relevant since, as demonstrated in the previous chapters, not all use cases may be applied to all the services. When creating particular solutions that comply with the GDPR, we may miss situations or give too much information regarding the regulation, overwhelming those working on the project that may not be too familiarized with it.

Each use case is accompanied by the GDPR principles they address and, for some of the use cases, the articles that discuss them. People who are aware of the regulation already know what is being discussed, but for those who do not, the description gives tips, requirements, and articles from which they can study more of what the regulation requires. This kind of structure helps those who may be confused about what the GDPR states and requests.

Most of the patterns were retrieved from an online source, while a few were adapted from [19]. The big difference between this library and *privacypatterns.com* is that, since the patterns are in the domain of privacy by design, they were not organized to match the GDPR problems. Many privacy concerns are notable but are not relevant to the GDPR case, so having them organized this way facilitates the search for solutions. We have more than one pattern that addresses the same requirements for most use cases but proposes slightly different approaches and solutions according to different situations or environments. This gives the developer more to choose from and choose what suits the service the best since some patterns may be better for one product and not another. Using patterns also gives the ability to adapt the proposed solution to the architecture that the organization aims to build. So even if the patterns cannot be fully applied, the ideas and steps required may help developers create something that is GDPR compliant while maintaining the original idea.

The diagrams of the use cases and the patterns are also something different, as previously mentioned. Even if the patterns cannot be implemented to their fullest, the diagrams help see what processes and architecture requirements the services need to provide to their users (the data subjects) to be GDPR compliant.

6.2. GDPR Principles addressed

This work practically covers all the requirements and principles of the GDPR and connects them, something that was lacking in the other works. Looking at Table 2 and Table 3, we can see that the principles and requirements there are and see that most of them are explicitly addressed in the use cases and, consequently, in the patterns.

Purpose limitation principle is addressed in the Register in system, Request for erasure of data, and Change of data processing purpose use cases. Data minimization in Register in system, and Request for restriction on personal data's processing use cases. The principles of trueness and accuracy were handled in Request for restriction on personal data's processing and Request for/and update of data use cases.

Integrity and confidentiality are in Register in system, Inform of Breach, Change of data processing purpose and Consent of Minors. Lastly, the principles of lawfulness, fairness, and transparency are addressed in most use cases except the ones regarding registration and the request for the update of the data.

In terms of the requirements mentioned, the data minimization is already specified. System security and privacy are handled in the Inform of Breach, and the Consent control is handled in all the use cases that require a review and writing of consent.

Data traceability considers logs, and these are addressed in many use cases. User access is present in the use cases related to requests, and the requirements of data rectification and data restriction are addressed on the use cases with the same name. The physical location of data has solutions in some patterns of the use case Register in system.

Regarding the principle of storage limitation, it is not expressly associated with any of the use cases. [19], proposes a data lifecycle to regulate how long the data can be kept and then provide an automatic erasure of the data; however, it states that it "is particularly difficult to achieve." The technical solution presented does not provide a concrete solution to this problem. This principle focuses on determining a period in which the data can be kept and processed. This analysis must be done while the data and the consents are being defined and written, as it must be shown to the user at the request of consent. Adding a pattern just for this reason would be too simple and become somewhat irrelevant. Also, although this principle is not addressed and handled in the use cases, it is present in the Guidelines for GDPR Compliance, section 4.2. These guidelines should accompany the library to introduce the requirements and the steps needed to achieve compliance with the regulation.

Although only noted in one use, the principle of accountability is somewhat the base for this work. All these solutions help the controller show responsibility in compliance with the GDPR. The use of logs is also an excellent way to demonstrate compliance.

6.3. Quality of the Library

We have already seen what this work has that is different from other works and how many of the discussed GDPR principles it addresses, so now we have to discuss the library's quality.

Most of the patterns used in this library are from a reliable source in the Privacy by Design domain, *privacypatterns.org*, and some are already used in many real-life services. These patterns already provide some assurance to the library's reliability since its application can be found in many services. The patterns that were adapted, Technical Solutions, have the base of the solution the same as the source. The changes were made by studying the GDPR, and context was created to fit the designed template.

6.4. Possible Benefits

We already have many patterns available to solve GDPR compliance problems, but they are not organized to fit the questions that come from the regulation. When the regulation was enacted, it took some time for companies and other organizations to adapt their services to comply with it. Although not the same instrument, other directives that focused on protecting personal data were already in place. However, since there were some differences, it was challenging to change already established architectures and services. The growth of collected data is still increasing, and new laws or small changes to existing legislation can appear, so if we keep creating new patterns to comply with these regulations, it is a non-stopping work.

This library uses solutions that already existed. Besides presenting them for those who did not implement them before, it can show the ones that did precisely in the GDPR it is complying to. If new changes arise, patterns can be added and removed, as well as the use cases. The fact that it has both plain text and diagrams gives two different explanations on the matter. Those who comprehend the diagrams can already know what the pattern requires, and those who do not or want to investigate more can read the context and solution.

Since a guideline was made to accompany the library, those who are not familiar with the GDPR can more easily see what needs to be ensured and propose changes that will ensure compliance with the regulation. Also, having diagrams for the use cases and not only for the patterns shows some of the requirements needed to comply with the GDPR without having to see all the patterns.

6.5. Shortcomings

There are still shortcomings and downsides of this library. Although the library can be organized differently according to new legislation or new needs, it still requires searching and creating new relevant patterns and modeling those patterns.

Secondly, the use of ArchiMate may be apparent for those who work in information technology (IT) departments, but other departments may have a more challenging time deciphering them. The guidelines here can help, but it is also essential to assess the compliance with the GDPR since it is required to demonstrate it, and this work is probably done by other departments that are not IT.

7. Conclusion

Data protection is important and crucial in a business, especially when personal data is stored and processed. The creation of GDPR confirms it. We live in an era where our data is easily acquired and processed without the owners' knowledge and sometimes without their consent. Luckily the regulation gives guidelines and rules for the organizations that operate in the EU to follow. The challenge is that there is much information and constraints to follow, and the language is not very explicit nor give objective rules to follow. This research contributes to ensuring Information Systems compliance to the GDPR, presenting ways of achieving it, using a library of patterns. When creating this library, the description and modeling of use cases were performed, and the definition of the associated entities and GDPR principles. A search through the sources was conducted to select the patterns that better solve the problems that the GDPR requirements bring to the use cases, and when needed, new patterns were created. In total, twenty-two patterns compose the library. This collection of patterns is used in the case study, demonstrating how services that require personal data processing may use the proposed solution and what changes when the patterns are applied. The guidelines also alert companies on what to consider if their services are processing personal data. Although very important in the design phase of a project, these concerns are permanent throughout its lifecycle. To point out that data processing occurs not only for users but also for the company's employees.

7.1. Contributions

This library provides eleven use cases and twenty-two patterns that help solve problems related to the GDPR principles and the data subject’s rights. Privacy by Design already had patterns that addressed these concerns but were not organized according to the regulation.

The template used is the following:

- Associated Use Case
- Associated GDPR principles
- Name (of the pattern)
- Context
- Problem
- Solution
- Source

This template uses the base elements of patterns’ templates, elements from *privacypatterns.org*, and elements that address the use cases. After each use case and pattern diagrams in ArchiMate showing the different processes and possible architectures are present.

This organization of the library is something that did not exist before. Much work has been done in the Privacy by Design domain and to solve GDPR compliance problems, but the connection between the two was not something very used before. Besides relating GDPR principles and Privacy by Design strategies, this library intertwines the GDPR principles with the data subject’s rights, which were usually addressed separately. GDPR’s principles and rights are tackled in this work, and even when no solution is present for some of the principles, there is an awareness to comply with them.

This library uses patterns retrieved from known sources, presents them with diagrams, and relates them to the GDPR, giving reliability to the library. The use of previous work prevents that every time some regulation or law is updated, new patterns have to be created; it just has to be found a new pattern that better address the new constraints.

7.2. Future work

In the future, we expect to add other patterns to the library, especially to the use cases where the patterns were hard to retrieve. Additionally, an interface could be created to show the collection of the use cases and patterns in a more dynamic way. Another future path to explore is developing a library focused on use cases for inner-company problems since the employees are also data subjects. With this, other concerns appear since the processing of personal data may not require consent due to contractual reasons. Another work that can be performed is addressing other architectural levels of the patterns or develop diagrams more focused on these levels, such as informational, applicational, and technological.

The GDPR is a new challenge that, at first glance, may seem hard to follow, but this can be solved with simple and accessible frameworks, interfaces, or guides that explain how companies can comply with the regulation. We can see that the GDPR brings a sense of security to the services, not only for the data subject but also for those who process data. Work on this domain is much needed and applicable to real-world needs and concerns.

8. References

[1] C. Alexander, S. Ishikawa and M. Silverstein, “A Pattern Language: Towns, Buildings, Construction”, Oxford University Press, 1977.

[2] L. Moné, "How to Solve GDPR with Enterprise Architecture: A Case Study," LeanIX, 24 5 2018. [Online]. Available: https://www.leanix.net/en/blog/how-to-solve-gdpr-with-enterprise-architecture. [Accessed 24 11 2020].

[3] "General Data Protection Regulation (GDPR)," Intersoft Consulting, n.d.. [Online]. Available: https://gdpr-info.eu/. [Accessed 24 11 2020].

[4] EU, "The History of the General Data Protection Regulation," European Data Protection Supervisor, n.d.. [Online]. Available: https://edps.europa.eu/data-protection/data-protection/legislation/history-general-data-protection-regulation_en. [Accessed 24 11 2020].

[5] R. Verheijen, "EXIN: Privacy & Data Protection, Whitepaper: Data Protection: Compliance is a Top - Level Sport," EXIN and Secura, 2017.

[6] A. Cavoukian, "Privacy by Design The 7 Foundational Principles, Implementation and Mapping of Fair Information Practices," Ontario, Canada, 2010.

[7] M. Colesky, J. Hoepman and C. Hillen, "A Critical Analysis of Privacy Design Strategies," in *2016 IEEE Security and Privacy Workshops (SPW)*, San Jose, CA, 2016.

[8] C. Alexander, "The timeless way of building", New York: Oxford University Press, 1979.

[9] F. Buschmann, K. Henney and D. Schmidt, "Pattern-Oriented Software Architecture: A Pattern Language for Distributed Computing" Vol.4, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007.

[10] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad and M. Stal, "Pattern-Oriented Software Architecture - Volume 1: A System of Patterns", Wiley Publishing., 1996.

[11] IT Governance Europe Ltd., "General Data Protection Regulation A compliance guide, Green Paper," IT Governance Europe, 2019.

[12] M. Lankhorst, "8 Steps Enterprise Architects Can Take to Deal with GDPR," 30 01 2017. [Online]. Available: https://bizzdesign.com/blog/8-steps-enterprise-architects-can-take-to-deal-with-gdpr/. [Accessed 24 11 2020].

[13] B. Studsgarth, "GDPR compliance assessment for SMEs," in *M.S. thesis, Governance and Strategies* , Aalborg University, Copenhagen, 2018.

[14] N. Doty and M. Gupta, "Privacy Design Patterns and Anti-Patterns," UC Berkeley, School of Information, California, 2013.

[15] J. Lenhard, L. Fritsch and S. Herol, "A Literature Study on Privacy Patterns Research," in *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Vienna, 2017.

[16] LeanIX GmbH, "Mastering the GDPR with Enterprise Architecture," LeanIX, Bonn, Germany, n.d..

[17] PDP4E Project, "PDP4E," n.d.. [Online]. Available: https://www.pdp4e-project.eu/. [Accessed 24 11 2020].

[18] K. Hjerppe, J. Ruohonen and V. Leppänen, "The General Data Protection Regulation: Requirements, Architectures, and Constraints," in *2019 IEEE 27th International Requirements Engineering Conference (RE)*, Jeju Island, Korea (South), 2019.

[19] D. Rösch, T. Schuster, L. Waidelich and S. Alpers, "Privacy Control Patterns for Compliant Application of GDPR," in *25th Americas Conference on Information Systems*, Cancun, 2019.

[20] A. Calabró, S. Daoudagh and E. Marchetti , "Integrating Access Control and Business Process for GDPR Compliance: A Preliminary Study," ITASEC, Pisa, Italy, 2019.

[21] C. Palmér, "Modelling EU DIRECTIVE 2016/680 using Enterprise Architecture," KTH, School of Electrica l Engineering (EES), Stockholm, Sverige, 2017.

[22] N. Doty and et al., "privacypatterns.org," UC Berkeley, School of Information, n.d.. [Online]. Available: https://privacypatterns.org. [Accessed 29 12 2020].