[TNOTE: Number]

Guidelines for GDPR compliance in Big Data systems[⋆]

Mouna Rhahla[a,b,*], Sahar Allegue[a,b,*] and Takoua Abdellatif[a,c,*]

[a]University of Carthage, Polytechnic School of Tunisia, SERCOM Lab, Tunisia
[b]Proxym-Lab, Proxym-IT, Sousse Tunisia
[c]ENISo, University of Sousse, Tunisia

ARTICLE INFO

Keywords:
The General Data Protection Regulation (GDPR)
Big Data analytics
Privacy
Security

ABSTRACT

The implementation of the GDPR that aims at protecting European citizens' privacy is still a real challenge. In particular, in Big Data systems where data are voluminous and heterogeneous, it is hard to track data evolution through its complex life cycle ranging from collection, ingestion, storage and analytics. In this context, from 2016 to 2021 research has been conducted and several security tools designed. However, they are either specific to particular applications or address partially the regulation articles. To identify the covered parts, the missed ones and the necessary metrics for comparing different works, we propose a framework for GDPR compliance. The framework identifies the main components for the regulation implementation by mapping requirements aligned with GDPR's provisions to IT design requirements. Based on this framework, we compare the main GDPR solutions in the Big Data domain and we propose a guideline for GDPR verification and implementation in Big Data systems.

1. Introduction

The General Data Protection Regulation GDPR [34] sets new requirements on security and data protection through 99 articles and 173 recitals and aims to protect the rights and freedom of natural persons. Every organization that deals with personal data has to comply with GDPR to protect these rights and to be accountable while improving business models [57]. Accountability aims at demonstrating how controllers comply with data protection principles. Each organization must answer the following questions: what information is processed? why? how and where is data stored? who can access it and why? is it up-to-date and accurate? how long will you keep it for? how will it be safeguarded and how accountability should be reached?

Some previous works present how to extract technical requirements from law requirements [2] before the birth of GDPR. Nevertheless, no design patterns or best practices can be directly applied in the Big Data context for GDPR-compliance implementation.

In the last few years, topics about GDPR have been discussed across a range of academic publications and industry papers from different theoretical and practical perspectives, including numerous implementations and design concepts for GDPR compliance [30]. These works are still in their infancy with a limited scope. Fully developed and approved tools that implement GDPR articles are still missing especially in the area of Big Data analytics. The term "Big Data analytics" refers to the entire data management life cycle from ingestion and storage to analysis of high volumes of data with heterogeneous format from different sources. As presented in Figure 1, the reference architecture of Big Data systems covers 5 main layers [67]: data sources, ingestion, processing, storage, distribution and services. At the processing layer, sophisticated algorithms are being developed to analyze a large amount of data to gain valuable insights for accurate decision-making, detecting unprecedented opportunities such as finding meaningful patterns, presuming situations, predicting and inferring behaviors. Due to the large data volume and the complexity of processing, tracking data dependencies and privacy verification are challenging. For this purpose, data security and governance layer is a cross-layer generally used for data security and management. Consequently, it represents a key part of the system in implementing GDPR requirements.

[⋆]This project is carried out under the MOBIDOC scheme, funded by the EU through the EMORI program and managed by the ANPR.

✉ Mouna.rhahla@proxym-it.com (M. Rhahla); Sahar.allegue@proxym-it.com (S. Allegue); Takoua.Abdellatif@ept.rnu.tn (T. Abdellatif)

https://www.proxym-group.com (M. Rhahla); https://www.proxym-group.com (S. Allegue); (T. Abdellatif)
ORCID(s):

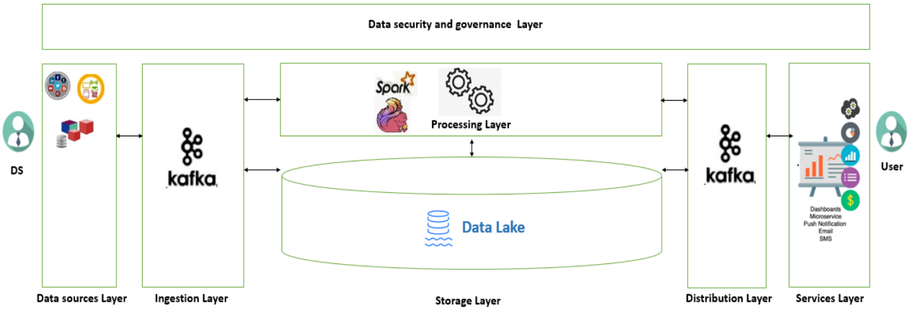


Figure 1: A classical Big Data system following a reference architecture [67]

Recent academic and industrial tools [18, 42] implement some GDPR requirements translating automatically the privacy policies to software in order to provide accountability. However, these works address, only partially, GDPR principles and their related articles such as purpose limitation, data minimisation, storage limitation, transparency or security [37, 60]. Other works concentrate on particular articles of the regulation [39] like the right to data portability, the right to be forgotten, the access right or the right to be informed [29]. Also, these works generally address one particular type of data source (logs, IoT sensors or classical SQL databases). It is not clear how to apply proposed solutions that consider uniform data, to Big Data architectures with multi-channel data sources, different purposes and intensive processing. Consequently, we still lack guidelines to verify GDPR compliance and to implement the regulation in a Big Data context. As a starting point, in order to address this issue, a comprehensive overview of the regulation and a common understanding of its key concepts are necessary. Afterward, the analysis of GDPR documentation and the study of recent works on privacy and GDPR allow the identification of the main privacy requirements and building blocks for GDPR compliance verification. As an outcome of this study, and based on the different experimentation carried out in the state of the art, we propose a framework with well-defined components to implement the regulation. According to these components, we situate the different works carried out on GDPR in the domain of Big Data. Furthermore, we provide an overview of how to use the framework to assist IT developers and Big Data system designers to build GDPR-compliant systems and applications. As an illustration of the framework usage, we consider the example of an e-health application, and we illustrate how we used the framework to help privacy by design implementation in the considered application. This paper's contribution can be summarized as follows:

- An analysis of GDPR principles and entities for a better understanding of the regulation by IT developers and Big Data system designers.
- A translation from GDPR principles requirements to IT design requirements.
- A framework for GDPR compliance verification and implementation in Big Data systems.
- A classification of the state of the art conducted on GDPR solutions implemented between 2016 and 2021 in both academic and industrial areas.
- A use case demonstrating the framework usage.

This work is an extended version of our previous work [62] which was restricted to a survey and a first version of the proposed framework. In this paper, we propose a translation from the regulation's requirements to IT design requirements which allows us to have a more precise and fine-grained framework. Furthermore, an IoT use-case is proposed to illustrate the framework usage that helps us identify missing parts in the use-case management system. Furthermore, we extended the related works' section and the GDPR tools section with recent solutions mainly from the industry. The up-to-date version of the studied solutions allows us to provide some key guidelines for GDPR implementation. Finally, the evaluation of the ameliorated solution shows an acceptable overhead when implementing GDPR-compliance.

This paper is structured as follows. Section 2 is an overview of GDPR principles and main entities. In section 3, we presented the related works and we highlighted the contribution of this paper. Section 4 presents the problem statement and illustrates the main GDPR challenges in Big Data systems. In section 5, we extract the main IT design requirements starting from GDPR principles and we describe our framework for GDPR compliance in Big data systems. In section 6, we use the presented framework to classify GDPR tools for reuse purposes. We describe the framework used for GDPR-compliance implementation and evaluation in section 7. Finally, section 8 provides a summary of the main findings of this paper and highlights new opportunities for future work.

2. GDPR entities and principles

The GDPR aims at delivering harmonized, consistent and high-level data protection across Europe. It has 99 articles and 173 recitals grouped into 11 chapters. In those chapters, it addresses a set of principles, entities, obligations and legal requirements. GDPR is a complex law and hard to understand and analyse by Big Data system designers and IT developers. In this section, we will illustrate a big picture of GDPR requirements and entities through top down approach.

2.1. GDPR entities

There are six main entities in the regulation [34]:

- **Data Subject (DS):** an identified or identifiable natural person, directly or indirectly, by data to be used by the controller or by any other natural or legal person. A data subject is any person whose personal data are being collected, held or processed.

- **Controller:** a natural or legal person, public authority, agency or any other body which alone or jointly with others determines the purposes, conditions and means of personal data processing. It ensures compliance with GDPR principles related to the processing of the personal data (Accountability), implements data protection policies and data security measures, carries out data protection impact assessment (DPIA) for high-risk processing, informs data subjects on their rights, in case of a personal data breach, notifies the supervisory authority within 72 hours and transfers personal data to the third country or international organization, per specific safeguarding provisions.

- **Processor:** (a person or a legal entity) processes personal data on behalf of the controller. Specifically, it collects personal data online through registration, contact forms, email or digital payments and invoicing. It also stores, uses, records, organizes, retrieves, discloses and deletes the collected personal data on behalf of, and under the instructions of the controller and creates inventories for all above-mentioned data processing categories.

- **Data Protection Officer (DPO):** (a person or a legal entity) manages and supervises all data protection activities. Specifically, he monitors compliance to GDPR personal data protection and security provisions and cooperates with the supervisory authority.

- **Supervisory Authority (SA):** Article 46 states that supervisory authorities "are responsible for monitoring the application of this Regulation and for contributing to its consistent application". The independent public authority is responsible for monitoring regulated entity compliance with GDPR.

- **Third party:** refers a natural or legal person, public authority, agency or body other than the data subject, controller, processor and persons who, under the direct authority of the controller or processor, are authorised to process personal data.

Dependencies between these entities are defined by GDPR articles: For example, the data subject can declare his consent to the controller (Art.4). He can also request from the controller access to data (Art.13 and Art.14), data rectification, make processing restriction and information about the life-cycle of his data. On the other hand, the controller provides information to data subjects (Art.13 and Art.14) and communicates data breaches to them (Art.34). The previous work [62] gives a big picture of the main articles' based relationships between the different entities. From this network, it becomes evident that the center of activity in this regulation revolves around the data subject and the main entity in the GDPR is the controller. A large set of actions is necessary. From a technical perspective, systems should be able to provide options for storing and revoking consent, as well as to restrict processing on a fine-grained level (Art.4, Art.21, and Art.18). The ability to deliver complete and coherent data to data subjects or transfer it to competitors has to be implemented (Art.20). The right to data rectification or deletion (Art.16 and Art.17) poses further challenges. In the next section, we will present the big picture of GDPR requirements summarised in its principles.

2.2. GDPR principles

GDPR sets out seven key principles for the processing of personal data stipulated in Article 5 [34]. They can be concluded in the following points:

- **Lawfulness, fairness and transparency:** Personal data shall be processed lawfully, fairly and in a transparent manner with the data subject.
- **Purpose limitation:** This principle aims to make clear that personal data should be collected for specified, explicit and legitimate purposes and not further processed in incompatible manner with those purposes. Controllers should define and document the purpose for data usage and provide the possibility to update purposes and to check the coherence between them.
- **Data minimisation:** This principle makes clear that personal data should be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed. In other words, the minimum amount of personal data is revealed to satisfy the application's purpose.
- **Accuracy:** Controllers should ensure the accuracy of any personal data created or updated with the right to rectification, which gives data subjects the right to have incorrect personal data rectified.
- **Storage limitation:** Controllers should consider which data to store, why and for how long. So, even if they collect and use personal data fairly and lawfully, they cannot keep it for longer than needed. GDPR does not set specific time limits for different types of data. Once information is no longer needed, personal data must be securely deleted.
- **Integrity and confidentiality:** Controllers must have appropriate security measures put in place to protect the personal data they hold and to have records to demonstrate compliance.
- **Accountability:** The accountability principle requires that controllers take the responsibility for what they do with personal data and how they comply with the other principles. They must have appropriate measures and records settled to be able to prove their compliance.

As we see, GDPR principles gives a big picture about the law requirements. It is really hard and complex to analyse the regulation and apply it to an IT product life cycle. In the next section, we will detail the proposed guideline scope.

2.3. Guidelines Scope

In this paper, our goal is to help Big data developers and systems architects to better address GDPR requirements for Big Data applications. The GDPR opts to having the "appropriate technical and organisational measures" established when processing personal data. These appropriate technical and organisational measures vary and depend on the use case and scenarios. In this work, we focus on the technical measures for systems adopting a classical Big Data architecture as presented in Figure.1. We followed a top-down analysis of the GDPR. We start from the general description of article 5 that refers to many articles and recitals as summarised in Table 1. For example:
"**Lawfulness, fairness and transparency**" are some of the main goals of the GDPR. Important Articles include Art. 5(1)(a) as well as the obligation to provide information stipulated in Articles 12 through 14. A separate Recital about transparency (Recital 58 and 39(1-4)) provides some technical recommendations and other articles and recitals describe the transparency principle requirements as presented in Table 1.
"**Purpose limitation**" is requested particularly through Art.5(1)(b) in accordance with Recital 39(6). Many articles and recitals are linked to the purpose limitation such as Art.6 and recital 50.
The main contributors to "**data minimization**" is Art. 5(1)(c) in accordance with Recital 39(7-8).
"**Accuracy**" is illustrated in article 5(1)(d) but it is linked to data subject rights since personal data should be "accurate and, where necessary, kept up to date. Every reasonable step must be taken to ensure that inaccurate personal data, having regard to the purposes for which they are processed, are erased or rectified without delay ". So requirements in Chapter 3 "Rights of the data subject", should be linked to this principle (articles and the corresponding recitals) such

GDPR principles	GDPR articles	GDPR recitals
Lawfulness, fairness and transparency	5(1)(a),6, 7, 12, 13, 14, 15, 19, 24(1), 25(1,2), 30, 32(1), 33, 34.	39(1-4), 42, 43, 58, 60, 61, 62, 63, 74(1-3), 78, 86, 87.
Purpose limitation	5(1)(b), 6(4) , 22, 24 (2).	29, 39(6), 39(9), 50(1), 71(1-2).
Data minimisation	5(1)(c), 25(2)	39(7-8) , 39(10), 78(2-3), 156.
Accuracy	5(1)(d), 16, 17, 18, 21, 13(2)(c), 14(2)(d), 15(1)(e)	39(11), 63(4), 65(1-4), 66, 67(1-3), 69(1-2), 71.
Storage limitation	32(1)(b,c,d), 5(1)(e), 13(2)(a), 20(1,2), 13(2)(a), 15(1)	49, 39(10), 63, 68(3-6), 83(3)
Integrity and confidentiality	5(1)(f), 28(3)(b), 32(1)(b), 25(2), 29, 32(2), 25(1), 32(1), 32(1)(b), 89(1), 25(1), 32(1)(a), 40(2)(d), 6(4)(f)	28, 29, 39(12), 49(1-2), 75, 78, 81, 83(2-3), 84, 88, 89, 90, 91, 95
Accountability	5(2), 24, 30, 32(1)(a,b,c), 34, 35(11), 37(7)	82

Table 1
GDPR requirements analysis

as Art. 16, Art. 17, Art. 18 as mentioned in the Table 1.

GDPR is a complex law and targets software compliance and also internal enterprise processes. In this work, we focus on the software compliance part whose articles and recitals are summarised in Table 1. We also limit the scope of our work on the consent as legal basis of processing. GDPR requirements of a contractual nature between its entities are not studied in this work because our target is IT developers. In addition, this work addresses a guideline for controllers. So, we are not addressing guidelines for processors obligations or for supervisory authorities obligation. Also, joint controller of Art. 29 and data transfer will be addressed in future work.

GDPR is a hard to understand law and very linked and dependent on its principles, articles, and recitals. We provided a summarised overview of the relationship on the dependency between principles, articles, and recitals to highlight that all GDPR articles and recitals are linked. That is why many organisations such as Cloudera [38] addressed GDPR compliance by analysing and studying its principles (adopting the same top-down approach). Also, academic works [63] followed some methodology to group GDPR articles by principles in their understanding of law.

3. Related works

Works on GDPR compliance can be divided into 3 main categories: (1) GDPR analysis, (2) Frameworks for GDPR compliance and (3) IT tools for GDPR implementation.

The first category of works presents theoretical interpretations of GDPR. The second category is more technical and presents some guidelines to implement GDPR compliant systems. The last category is the scope of Section 6 about recent implementations and tools of GDPR in Big Data. In this section, we focus on the two first categories. The third category that focuses on IT tools and implementations in Big Data systems is the scope of Section 6 of this paper.

3.1. GDPR analysis

Several authors and organizations have analyzed GDPR and privacy by design. However, they generally provide documentation and support for law understanding without providing a practical framework or guidelines to apply this law in the company's projects or products as in [77].

Other works generally focus on specific applications such as healthcare [46, 54, 79], private bank [69] or specific tasks like data storage [70]. They analyze and discuss the impact of GDPR compliance on the specific fields of interest. For example, in [54] the fundamental legal issues are identified, as well as challenges and opportunities for an e-health scenario. Architectural guidelines are also proposed and tackled without a proof of concept. In [41], authors analyze GDPR articles and propose an ontological data protection model for an organization. In [44], authors highlighted the effects of GDPR on designing assistive environments without proposing implementations. In [47] and in [68], authors analyze the principle of data protection by design without proposing implementation guidelines. In [21], authors provide guidelines for implementing GDPR from a user experience perspective for subsidiary companies, but no implementation or use case validation is proposed. In [51], authors present an overview of the GDPR in terms of entities involved and provide a systematic representation of their interactions. Consequently, the paper presents an analysis of the entities categorised according to their role as defined by GDPR, the nature of information flows between these entities, and the requirements for their interoperability. These studies helped us to understand more clearly GDPR challenges and its requirements. Also, it helped in our translation from legal requirements to technical requirements [63].

3.2. Frameworks for GDPR compliance

A framework is proposed by the Special privacy project in [18] and provides a set of functionalities for consent, transparency and compliance checking. This work illustrates the approach to consent management and consent implementation in a big data context. Therefore, authors studied the transparency requirements to GDPR and provided a compliance and transparency dashboard. In our work, we addressed not only consent and transparency but also other GDPR principles like accountability, data minimisation and purpose limitation. Regarding consent, we provide a more fine-grained consent policies based on attribute-based access control.

In [55], authors provide guidelines for managing consent and personal data in ICT businesses, taking into account the provisions of the General Data Protection Regulation (GDPR). They started from the analysis of previous studies on consent management models and GDPR requirements. In our work, our starting point is the GDPR law with its principles, articles and recitals and then, we extracted the IT requirements as guidelines for IT developers. Furthermore, in [55], no implementation or evaluation is provided, the result of the study can just help data controllers to improve the integration of consent and data management in their systems.

Another project, called PrivacyGuide [75] is based on machine learning and natural language processing techniques to classify privacy policy content and to calculate the risk behind each policy. A visualized summary is provided to illustrate the relevant privacy aspects associated with the identified level of risk. This work helps identifying risk assessment and accountability principles. In our work, we provide an assessment and a risk manager as an IT component of our framework. Its role is to compare Business policies against consent policies for non-compliance detection in a static way. Inspired by PrivacyGuide project's ideas, this module can be enhanced for a dynamic privacy risk analysis.

A framework for GDPR compliance for SME (Small and Medium-sized Enterprises) is proposed in [20]. The framework distinguishes 3 phases to follow: analysis, design and implementation. The design phase of the project focuses on three principles: routines, policies and templates. Work routines concern the handling of personal data during a regular working day for all employees. Starting from work routines and the SME policies, a template is generated and is displayed in a GUI. In [20], authors actually provide a very general framework that may cover different SME. Giving the presented framework, organisations need to work with experts in order to adapt it to its routines. This work can be applied to Big Data systems for data flow control. However, for GDPR implementation, the proposed solution is very general, and additional work is required with experts and IT design systems.

In [16], authors propose a framework for the GDPR-compliance in smart city IoT platforms. The adopted approach is different from ours. Indeed, instead of starting from GDPR principles' study to deduce IT requirements, authors identify a set of security and privacy requirements related to IoT systems matching then the list of requirements against GDPR features extracted from GDPR principles. In our work, our starting point is the GDPR articles and recitals and then we extract IT requirements. The proposed framework is guided by the IoT architecture and the focus is put on securing M2M communications. Compared to their framework, ours is guided by the particular reference architecture of Big Data systems which has many similarities with IoT. Indeed, we have in common the management components related to data collection, storage and processing. In our framework, we additionally add a data flow manager that represents a global view of data flow stages (sources, edges, processors, etc) which is needed for accountability. We also adopt a more fine-grained architecture where risk management, distribution management, user consent, storage and processing are handled in separate components. This provides more flexibility in managing GDPR principles. Furthermore, we show, in this paper, that our framework can be directly applied to IoT as a particular use case of Big Data.

GDPR-compliance is currently handled in a lot of EU projects, namely BPR4GDPR [19], DEFeND [32], SMOOTH [72], PDP4E [53], PAPAYA [52] and PoSeID-on [59]. These projects address the lack of specific, operational solutions that respond to challenges and legal innovations posed by GDPR, by providing systematic methods, detailed techniques and software tools [24].

The goal of BPR4GDPR [19] (Business Process Re-engineering and functional toolkit for GDPR compliance) is to

provide a holistic framework able to support end-to-end GDPR-compliant intra- and inter-organisational ICT-enabled processes at various scales. The proposed solutions are very general and aim at covering the full process lifecycle, from its initial identification or specification to its enactment and execution. They target diverse application domains while our work focuses on the specific domain of Big Data systems.
DEFeND [32] is the data governance framework designed to assist organizations to implement GDPR. DEFeND2 adopts the same approach as ours where project members start from the study of GDPR articles and principles. Then, they identify the framework components implementing these principles. However, their scope is Larger than ours since they target coordinators from different sectors (e-health, banking, energy etc) and they propose solutions for both operations and for technical implementations [76] [58]. If we compare the implementation module of DEFeND project to our framework, we see that we are more guided by the Big Data architecture and then we provide a more fine-grained architecture. For instance, since we have a data collection layer different from the processing and storage layers in Big Data, we propose three managers for each layer, the collection manager, the processing manager and the storage manager to deal with each layer separately.
As we can see, many projects are seeing the light in the area of GDPR. These projects are with different objectives and different scopes. As it is not yet feasible to address all the issues, each project has its own target public and/or focuses on some specific aspects of GDPR. Although the work is carried out, several other aspects of GDPR remain still open such as purpose limitation, data minimization and storage limitation [24].

To summarize, compared to existent projects and proposed frameworks for GDPR-compliance and to the best of our knowledge, our work presents the first technical GDPR framework and IT guidelines for GDPR compliance in Big Data systems. Although some ideas and design components are similar to some proposed solutions, our framework is more adapted to the data touch points management and to the layers of the Big Data architecture. We also provide a set of technical implementations full-filling Big Data challenges and we illustrate the framework usage through an IoT use-case.

4. Problem statement

In this section, we discuss the major GDPR challenges in Big Data systems [33]. From our perspective, the current GDPR and privacy challenges can be grouped into four categories following the architecture layers presented in Figure 1 as follows:

- **Challenges in Data Sources layer:** Regarding the privacy principles, both consent and purpose limitation principle must be considered before beginning the collection phase. Each data subject has the right to know the reasons behind collecting each data from the sources. Hence, the data subject is asked to set his preferences about the collection frequency, the data granularity and the set of information he allows to disclose to third party applications. Preserving privacy at the source layer is essential and can affect the whole data life cycle. One may see that the principle of “data minimisation” and Big Data are at first sight contradictory and very challenging because the perceived opportunities in Big Data provide incentives to collect as much data as possible and to retain this data as long as possible for yet unidentified future purposes.

- **Challenges in Data Ingestion layer:** Big Data applications typically tend to collect data from diverse sources and without careful verification of the relevance or accuracy of the data thus collected. This can provide false analysis results and affect data quality. For example, in the field of e-health, incorrect data about the patient health or environment can lead to an erroneous diagnosis which puts the life of the data subject in danger. Anomaly detection systems are required for accuracy. Indeed, it is important to make sure that data are not modified during the transmission phase, and malicious entities that try to inject data in order to congest the network or influence the analysis results are detected.

- **Challenges in Data Processing and Storage layers:** Data are stored and processed to provide advanced and calculated information for the services layer. However, personal data should be stored for a well-defined time duration (Storage limitation). Thus, data retention and disclosure limitation are required at this phase. Consequently, necessary mechanisms must be deployed for destroying data when expired. Furthermore, a lot of data are collected for non-defined purposes mainly for Big Data analytics that requires the maximum of input to improve algorithms’ accuracy. However, the blunt statement that data are collected for any possible analytics

is not a sufficiently accepted purpose. The principle of storage limitation may undermine the ability to be predictive, which is one of the opportunities rendered possible by Big Data analytics. Indeed, if Big Data analytics allows predictability, it is precisely because algorithms can compare current data to stored past data in order to determine what is going to happen in the future.
Another challenging issue for securing a Big Data system is data sharing. For example, road traffic data can be collected by deployed cameras or travelers' smartphones and GPS in a crowd-sourcing way. During global road planning, it is challenging to define the access policy and enable privacy-preserving data sharing among the involved applications and services. Therefore, Big data storage and sharing require the deployment of appropriate techniques in order to respect the user consent and privacy while providing innovative analytic processing for different purposes.
Once a Big Data application resolved all the previous challenges for the processing and storage layer, a controller needs to demonstrate this. Here comes the role of transparency and accountability. The controller needs to provide all information for the processed data by providing where data are stored and how they are manipulated or processed. This task can be easy for classical applications. But in Big Data context, it is a challenging task. Big Data processing is complex and with different purposes and intensive processing and some time with opaque processing operations. Building, transparency and tracking data usage and storage is very difficult.

- **Challenges in Distribution and Services layers:** Third-party applications have access to the analytic results calculated from citizen data. The communicated information, even anonymous, may reveal personal data. It is important that data sharing be controlled with regard to citizens' consent. The challenge exists when there is a big number and a diversity of applications using personal data and communicating from data sources to processing layers. In that case, tracking data access and data breach notification becomes difficult. Also, in the context of Big Data analytics, the processing can be opaque whereas individuals (data subject) must be given clear information on what data are processed. They have to be better informed on how and for what purposes their information is used and in some cases, they require the logic used in algorithms to determine assumptions and predictions about them.

From the above description, the core GDPR principles seem, for the most part, to contradict some of the key features of Big data application and Big Data analytics. Nevertheless, rethinking some processing activities and IT developments may help to respect privacy, notably by having well-managed, up-to-date and relevant data while preserving Big Data spirit. Ultimately, this may also improve data quality and thus contribute to the analytics. Addressing GDPR principles requires a coordinated strategy involving different organizational entities including legal, human resources, IT security and more. GDPR includes key requirements that directly impact the way organizations implement IT security. Unfortunately, it is not possible to buy a GDPR-compliant product and think a system is compliant. Because GDPR is more about security processes and managing risk, no product will solve all of the privacy problems. What is needed is to ensure that solutions work together to be truly GDPR compliant [50].
The next section will detail our steps and building blocks of the proposed framework. We translated GDPR requirement into technical requirements trying to face the different challenges in Big Data architecture and provide a GDPR compliant solution.

5. From GDPR principles to IT GDPR framework

To design GDPR-compliant systems, GDPR obligations have to be interpreted as technical requirements that are not straightforward. We need to represent a valid means to write simple and understandable requirements. Some academics started to address this step as soon as GDPR appeared such as in [64] and in [43], the study of these similar efforts helped us in identifying and confirming the right IT requirements. It is the scope of this section where we follow the steps presented in Figure 2. Indeed, GDPR principles are interpreted and detailed as GDPR requirements that we translate to IT design requirements. Table 2 shows 12 requirements (**Req 1 - Req 12**) detailing the GDPR principles. These requirements are built based on the articles and recitals obligation related to each principles in Table 1. For example, the lawfulness, fairness and transparency principle leads to at least four requirements. First data subject (DS) privacy preferences have to be collected and applied when processing and communicating his data. This is the scope of the first requirement about "DS consent management". In particular, when a data breach occurs, DS must be notified (Req 2) and each time his data are used, data access has to be checked (Req 3) and communicated the DS (Req 4). In addition, in our translation to IT requirements, we not only consider GDPR requirement but also Big Data privacy

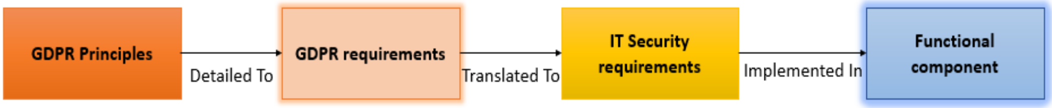


Figure 2: Steps for GDPR implementation

GDPR principles	GDPR requirements
Lawfulness, fairness and transparency	- **Req 1:** DS consent management. - **Req 2:** DS notifications in case of data breach. - **Req 3:** Data usage verification: data are used as expected. - **Req 4:** Data usage communication to the DS and continuous checking of lawful basis of data processing.
Purpose limitation	- **Req 5:** Purpose definition and documentation for data usage. - **Req 6:** Purpose update.
Data minimisation	- **Req 7:** Data collection limitation to the purpose.
Accuracy	- **Req 8:** Accuracy verification of any personal data created or updated and accuracy challenges and mistakes consideration. - **Req 9:** Dynamic verification of the data policy implementation.
Storage limitation	- **Req 10:** Consideration of which data to store, why and for how long.
Integrity and confidentiality	- **Req 11:** Security policies definition as integrity and confidentiality constraints.
Accountability	- **Req 12:** Compliance demonstration for DPO and DS: demonstrating principles' implementation.

Table 2
From GDPR principles to GDPR requirements

challenges layer by layer against GDPR. For example, in the ingestion layer, we are providing **Req 1, Req 2, Req 3, Req 4, Req 7 and Req 8**. Following the Big Data architecture life-cycle, we are providing requirements for every layer. We detailed in the next section, how we analysed Big Data architecture to detect data touch-points, and how our requirements control the pipeline and the data flow from collection to distribution.

The second step consists of extracting the main IT design requirements from the obtained GDPR requirements. We present hereafter the list of identified IT requirements.

- **IT Req 1: DS consent management.** A controller has to provide the API to the data subject DS in order to express his privacy preferences (definition of consent, withdrawal of consent, storage of consent, and compliance to consent). The DS has to be able to describe what data are being collected, the source, the reason of collection, the access rights to processors, where is data allowed to be stored, how long is the data retained, who has access to the data, to where and to whom is the data being transferred. Then, the controller translates each preference into a machine-readable policy language. Furthermore, the controller system must support signed consent to authorize the usage, access and management of Data Types. The concept of Data Type is derived from GDPR and can be regarded as Data Category. According to GDPR, the authorization/delegation to manage personal data (Types) provided by a user to the Big Data platform management must be performed by using a signed consent and the grant can be revoked any time by the user. So, when talking technically a consent is translated to a privacy policy with a set of conditions and constraints. Consent management refers to privacy policies management.
- **IT Req 2: Data usage verification: data are used as expected.** A controller has to map all relevant personal data and data flows to understand what to do with that data. In this phase, metadata about identified data elements

relevant for GDPR can be loaded into a data governance platform and classified then displayed, for instance, as a graph or dashboard. According to GDPR, users can delegate access rights to other parties. The delegation can be fine-grained by specifying data attributes. Attributes are about: data sets, devices, third-party applications, storage spaces, users, purposes.etc. For example, in e-health, a patient may be interested in granting access to a partner only about the glucose level and keeping the other information private. Therefore, only the owners or delegated users can access the data. On the other hand, a controller has to manage data ownership (permitting the change of ownership) and the access delegations. In the delegation management, it must be possible to list them (check the grants provided) and to revoke delegation or the consent.

- **IT Req 3: DS notifications in case of a data breach.** A controller has to ensure the communication of policy interference and breach to the DS and DPO in order to negotiate a modification of denying access. Real-time monitoring and notification management must be implemented. Also, a controller has to inform users about the security level at which the solution may work according to the level of security taken (it may depend on the kind of sensitive data managed). Furthermore, the GDPR requires the support of data breach detection in a short time whenever some data and Data Type have been tampered or leaked.

- **IT Req 4: Data usage communication to the DS and continuous checking of lawful basis of data processing.** A controller needs to implement an automated discovery of relevant or partial personal data. It also has to harvest metadata from heterogeneous solutions, data management, data warehouse, data integration, extract-transform-load, business intelligence, Big Data and Hadoop technologies [9]. This allows data identification through the exploration of connected metadata. In addition to using a privacy dashboard or navigable visualization so that the DS can explore and track his data flows between systems services. For example, the developers of Big Data applications need to create connections with: Dashboards (for presenting data and collecting actions from users or services), storage (for getting access to historical data, or for saving additional data, results of some data analytics) and with ingestion layers (brokers in case of IoT application) (for subscribing on the data drive or sending/receiving messages), etc. Big Data applications also invoke and implement Data Analytics processes exploiting a large amount of data storage, for example, by using machine learning approaches. Thus, the authentications to establish these connections have to be automated. This means that the developers are not forced to use the credentials in the source code to establish authenticated connections (for example, with the IoT Brokers, Dashboards, Storage, etc.). This can be implemented as an orchestration component facilitating this connection between the different components.

- **IT Req 5: Purpose definition and documentation for data usage.** A controller has to enrich and extend its access control policy with purposes for data usage to limit access to the specified policy. These policies need to be stored and evaluated with a policy engine for each data access.

- **IT Req 6: Purpose update.** A controller has to provide an update User Interface and flexibility to update defined policies for the defined purposes.If the controller performs some form of processing different from the one initially defined, the controller shall ensure that it is analysed, justified and documented why the new purpose is considered consistent with the old one. Updating privacy policy and/or revoking it means revoking the consent which is translated as privacy.

- **IT Req 7: Data collection limitation to the purpose.** A controller has to filter data at the ingestion layer (collection phase) based on the linked purposes for every data usage by using annotation techniques (data minimisation). When data are annotated with the appropriate purpose, they can be easily filtered. Anonymisation or pseudonymisation might be helpful in order to minimize the collection of excess data. Also, storage limitation requirements play an important role in minimizing data. (The controller may only collect and process data that are necessary for the defined and documented purpose. This includes each individual data attribute as well as the overall data set.)

- **IT Req 8: Accuracy Verification of any personal data created or updated and accuracy challenges and mistakes consideration.** A controller has to check data sources against the expected data sources defined in the consent policies. To provide data quality, we can use techniques such as trusted execution environment TEE [66] and remote attestation [40] to build trust in data sources. Personal data shall be accurate and, when necessary, kept up to date. Every reasonable step must be taken to ensure that inaccurate data, taking into consideration the purposes of the processing, is erased or rectified without delay. Here, the controller has to implement CRUD functionality for the DS dashboard (Data management: Create, Read, Update and Delete).This requirement refers to implementing the Rights of the data subject.

- **IT Req 9: Dynamic Verification of the data policy implementation.** A controller has to give the DS the ability to create or update his policies dynamically by accessing the policy storage and check interference problems. In this requirement, we need to implement appropriate algorithms for policy checking. The controller also has to respect the DS rights by providing the sharing, deleting, rectifying and calculating DS data as features.

- **IT Req 10: Consideration of which data to store, why and for how long.** A controller has to extend the policy implementation in order to define conditions for storage limitation. The controller has to give the DS the ability to express for a given data why it should be collected and how long it will be stored or processed. These policies need to be stored and evaluated with a policy engine for each data access. This shall provide the controller with the ability to identify any personal data no longer required. This helps the controller to identify and transform stored personal data no longer required into anonymous data. The anonymous data shall be in a format that prevents de-anonymisation with realistic effort. When data are anonymised, the system shall replace all copies of the original data by the anonymised data, unless they are deleted. In addition, the controller should have the ability to delete personal data identified as no longer required, including all instances of data such as backups.

- **IT Req 11: Security policies definition as integrity and confidentiality constraints.** A controller has to provide authentication and authorization techniques for processors, third-party users and IT personal. The controller can propose different protocols and modalities (push/pull) to authenticate and establish secure connections with third-party platforms or devices or users. For example, authentication can be based on certificate and/or access tokens are used, and then activate SSL/TLS connections supported by mutual authentications in the best cases Secure communications are required for all kinds of connections involving data source devices, ingestion Brokers, Big Data applications, dashboards and storage containers. A controller can also protect data by using adequate encryption algorithms such as attribute-based encryption ABE [65]. Besides, following the DS consent, the controller has to hide the identity by using a pseudonym and ensure a pseudonymous identity that can not be linked to a real identity during online interactions with third-party users and IT personal.

- **IT Req 12: Compliance demonstration for DPO and DS: demonstrating principles' implementation.** A controller has to monitor, block and audit by collecting, consolidating, securing, and analyzing audit logs. Tracking the risk assessment history highlights how much progress is being made over time. Also, audit logs are used to demonstrate accountability by comparing the processing scope against the consent policy defined by the DS. The controller has to support auditing for each data subject DS, to monitor who has accessed their personal data. The DS has to access the auditing data, obtaining details about the accesses, such as: when, where, how, and who accessed data. This feature is requested explicitly by the GDPR. The processing of personal data should be documented and this documentation be versioned and kept up to date. Also, continuous testing for software vulnerabilities should be performed and re-testing all security requirements for new releases as well as re-evaluating whether new security requirements may have evolved through progress in (state-of-the-art).

Overall, GDPR addresses the key security tenets of confidentiality, integrity, and availability of systems and data. Starting from this IT design requirement, we propose the framework for GDPR compliance implementation as technical components in the next section.

5.1. Framework architecture for GDPR compliance implementation in Big Data systems

To implement the main IT design requirements in a Big Data system for GDPR compliance, we analyzed the data life-cycle within a Big Data system and we highlighted the different data touchpoints by users or applications following the different layers of Figure 1. Our analysis resulted in five data touchpoints going from data sources layer to the distribution layer presented in Figure 3. Data touchpoints are a set of contact points with data by a user or by an application. For example, in the distribution layer, a service or an application from the services layer may ask access for the data. The touchpoint is the communication channel that has to be controlled so that only authorized data are disseminated to the application.

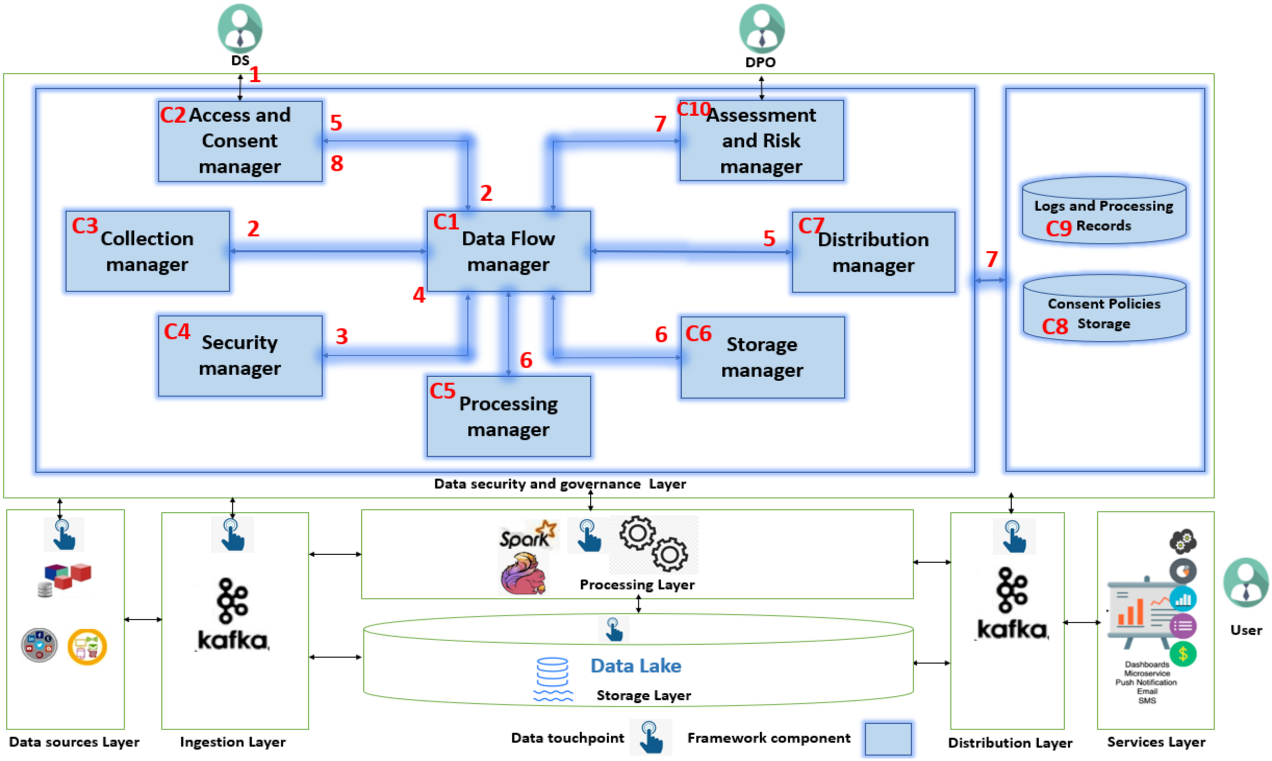


Figure 3: The GDPR framework architecture for Big Data applications

Figure 3 highlights the main functional components of our framework. The framework components are typically located in the data security and governance layer of the Big Data architecture and can be distributed following the workload application type. Each component communicates with the appropriate layer of the architecture and following data touchpoints distribution. These components are flexible, adaptable, and interact in a complementary way to provide compliance with the target system and workload. We identified a central component, the Data Flow Manager that plays the role of orchestrator to the other components. Furthermore, this component centralizes the global view of data as it flows in the Big Data system. Indeed, we noticed that data tracking and information control requires a global view of the data dissemination. Thus, graph-based representation of the system and data is very useful to control data access rights and to detect security breaches. For instance, we provide in Figure 3, a graphical representation of data dissemination in a big data system.

We present in this section the main components of the framework and then, we describe their operation and intercommunication.

- **C1: Data Flow manager.** With a centralized representation of data flow throughout the Big Data system, this component can manager data life-cycle (from collection to deletion) and detect security breaches. It relays on the other components (C2-C10) for specific tasks like notifying the DS in case of a data breach and keeping the DS informed about his data usage (**IT Req 3 and IT Req 4**).

- **C2: Access and Consent manager.** It implements four IT design requirements: **IT Req 1, IT Req 3, IT Req 5 and IT Req 10**. It provides the API to get the consent policies from the user and stores them in the **Consent Policies Storage C8** component. It also implements and enforces the GDPR DS rights: the right to be informed, to access, to erase, to update, the right of portability, processing restriction, and right to object.

- **C3: Collection manager.** It implements the following IT design requirements: **IT Req 2, IT Req 4, IT Req 7 and IT Req 8**. It is in charge of annotating raw data with consent policies defined in the access and consent manager. Data minimisation is enforced in this component by filtering the collected data at a very early stage.

- **C4: Security manager.** It implements **IT Req 11**. It defines security policies as integrity and confidentiality constraints. Also, it provides as explained **IT Req 11** pseudonymisation which is required by the GDPR. For this goal, the Big Data technology can be used like Apache Ranger [11] for pseudonymisation. Another technique is tokenization where data are permanently replaced by a substitute value, making the original data completely unrecoverable. Another solution is encrypting the identity of the DS using a robust algorithm such as ABE [65].

- **C5: Processing manager.** It implements **IT Req 2 and IT Req 3**. It helps controllers check the process scopes against consent policies defined in the consent manager component by the DS. Also, every processing activity is stored in the **Logs and Processing Records component C9**. The Logs and Processing Records component is a database.

- **C6: Storage manager.** It implements **IT Req 6, IT Req 8 and IT Req 10** requirements. It helps controllers manage the stored data (raw data and calculated data). This component allows the management of data location and Data Time To Live DTTL as a policy by checking regularly storage purpose, DTTL, and records updates. It controls data transfer by comparing data location defined in the consent policy and the current physical location of the data.

- **C7: Distribution manager.** It implements authentication and authorization techniques to control the user or service access to personal data (**IT Req 11**). It can provide also the finding data IT feature (**IT Req 4**) if the user is a data subject.

- **C10: Assessment and Risk manager.** It implements **IT Req 12**. It automates different functions of a privacy program, such as deploying Privacy Impact Assessment PIAs, locating risk gaps, demonstrating compliance and helping privacy officers scale complex tasks requiring spreadsheets, data entry and reporting. This component can be used by the data protection officer DPO as an audit service. For Example, it provides an output to the DPO GDPR-Compliance status and accountability information and demonstration. This is achieved by comparing the scope of consent policy collected from the DS against business policy concluded from the services process. Then, it reports a compliance status for each service and demonstrates compliance or non-compliance for users.

In addition to these components, two meta-data databases are generally required in this GDPR architecture design: **Logs and Processing Records C9** and **Consent policies storage C8**. These two storage points help to storing all processing records and logs in order to show compliance to the DPO and also to store the consent collected from the DS as security policies. These proposed components interact with each other, their relationship is detailed in the next section.

5.2. Framework operation and component interactions

In this section, we describe the different steps of the framework operation following the Big Data pipeline with its different data touchpoints. The framework consists of two important dashboards: (1) Consent management and transparency dashboard, which is responsible for obtaining consent from the data subject and (2) DPO transparency and compliance dashboard, which is responsible for presenting data processing and sharing events in an easily digestible manner and demonstrating that existing data processing and sharing complies with user policies. We present the steps performed by our solution starting from data sources layer to data services layers controlling the different data touchpoints as follows:

- **Action 1:** The DS uses a mobile application or a web application to access his Big Data application. During this setup, he passes by a human language consent request associated with a data usage policy. Before starting the collection of DS's data in the ingestion layer, a controller should obtain the DS consent.

- **Action 2:** C1 collects the signed consent by the DS and sends it to the Collection manager C3. C3 annotates collected data with adequate policies. Metadata management techniques can be typically used to extract the data flow as a graph representing data lineage. Afterward, collected data (node in the data flow graph) is labeled with the DS collected consent information as annotations or tags.

- **Action 3:** Inside C1 and based on the data flow graph, a policy checker verifies that each data annotation takes into account the preferences of the DS and that each processing or storage layer has access to authorized data only. This task is performed each time new data are received or whenever newly calculated data are obtained. If everything goes right in the configuration process, policies are enforced and stored in C8 (C1 sends policies to C8) else a notification is sent to the DS for consent rectification in C2.

- **Action 4-5:** After security checking, C1 sends the collected consent from C2 to C4 for integrity and confidentiality implementation. Following the annotation, data are either encrypted or private attributes annotated by "PII" are hidden (anonymized).
- **Action 6:** In the processing and storage layer,data lineage or data flow graph of all stored or processed data in this layer is captured. As presented in Figure 3, the DS's data are stored in a data lake and is processed in batch or real-time mode for the purpose defined in his signed consent. A unique token is generated for the DS's data and returned to him as a confirmation ID for his account creation in the application. Using the unique ID (or Token), he can access and control all his collected and processed data and his consent. He can also manage his rights as defined in the GDPR articles. C5 and C6 monitor and control the data storage and usage then return the result (as a data flow graph) to C1 to execute the configuration checking and return a notification if a problem exists to the DS or just display accountability information in the DS dashboard and DPO dashboard.
- **Actions 7:** In order to demonstrate compliance to the DPO, an Assessment and Risk manager dashboard is provided. A compliance checking is performed Here. C10 retrieves consent policies stored in C8 the Consent Policies Storage database and C9 the log and processing record for each data category.
- **Action 8:** Finally, a Transparency and Compliance dashboard is also provided to the DS. First, the transparency dashboard gives all details on the processing of the collected data in order to check that the DS consent is respected. If the DS decides to revoke the given consent and asks the controller to delete all of his data. The information stored in the storage points to the data he is referring to, hence all traces are automatically deleted, or he can just rectify his consent and here the policy configuration checker is activated by C1 with every change to keep policy implementation valid. In addition, the controller needs to provide a simple representation of the DS data flow graph and the propagation of his consent with fine granularity. First, C1 extracts the graph from C5 and C6 by the provided TrackToken and sends it to C2. Then, C2 redefines the data flow graph using a graph calculation framework. The DS has, clearly, a complete overview and full control of his collected data and consent.

In this section, we start from GDPR principles and obtained as output a well defined functional component for use. As we see, the framework components operations are valid for a wide range of Big Data applications and domains. Requirements are defined, we need now to provide an implementation technology for each component. In the next section, we will study the GDPR tools and detect which is the best implementation for every component.

6. IT tools for GDPR implementation

GDPR-oriented tools are divided into 3 main categories: (1) Academic GDPR Tools, (2) Industrial GDPR tools and (3) Apache tools that are built-in Big Data solutions. The next sections summarize these three categories.

6.1. Academic GDPR tools

In the past three years, many authors have worked to provide privacy tools for GDPR. These tools partially cover GDPR principles and articles:
In [70], the authors analyzed the impact of GDPR on storage systems and extracted security requirements and the adequate storage feature to provide GDPR compliance. Then, they took the case of Redis [15] to extend its feature in order to be GDPR compliant and measured the performance overhead of each modification. They found that achieving strict compliance efficiently is difficult. The authors highlighted three key challenges: efficient logging, efficient deletion and efficient metadata indexing for GDPR compliance. This work addresses three components of our framework: Storage manager, Security manager, and Collection manager.

Authors in [39] propose privacyTracker, a GDPR-compliant tool that covers data tractability and transparency. They implement some GDPR rights such as data portability and the right to erasure. A privacyTracker framework is an approach that empowers consumers with appropriate controls to trace the disclosure of data as collected by companies, and assess the integrity of this multi-handled data. This is accomplished by constructing a tree-like data structure of all entities that received the digital record, while maintaining references that allow traversal of the tree from any node, both in a top-down manner and bottom-up manner. A prototype was developed based on the privacyTracker principles as a proof-of-concept of the viability of the proposed principles. This work addresses Collection manager and Distribution manager.

For GDPR accountability in IoT systems, an IoT Databox model is proposed providing the mechanisms to build trust relations IoT [29]. The IoT Databox is an edge solution that implements the local control recommendation and collates personal data on a networked device situated at home. It meets the external accountability requirement by surfacing the interactions between connected devices and data processors and articulating the social actors and activities in which machine-to-machine interactions are embedded through a distinctive range of computational mechanisms. This model touches more than one component of our framework: Access and Consent manager, Collection manager and Distribution manager.

In a previous work [61], we propose a GDPR controller for IoT systems where security, transparency and purpose limitation are implemented. In this work, we start by providing three components: the Access and Consent manager, the Security manager and the Data Flow manager using Kafka topics. Also, in [4], we propose a security model for data privacy and an original solution where a GDPR consent manager is integrated using the Complex Event Processing (CEP) system [8] and following the edge computing. We show, through a smart home IoT system, the efficiency of our approach in terms of flexibility and scalability. We express policy in *5W* policy model: It is crucial for an individual to be sure that what he has shared is exactly what he wants to be shared, to whom, for what purpose and when. Individuals must have control over their data and can give or revoke permission to access their data for a given service whenever they want.

In [36], authors present TagUBig - Taming Your Big Data, a tool to control and improve transparency, privacy, availability and usability when users interact with applications. For IoT system, ADvoCATE [60] allows data subjects to easily control consents regarding access to their personal data. The proposed solution is based on Blockchain technology. Juan Camilo proposed another Blockchain-based solution to implement consent in GDPR [23]. This work provides the data subjects a tool to assert their rights and get control over their consents and personal data. These works are different implementations of the framework component using Blockchain technology.

In [35], the authors discussed how static program analysis can be applied to detect privacy violations in programs. The solution is based on classical information flow control techniques, tainting and backward slicing. Although important, the solution addresses a limited part of data control requirements in GDPR.

To the best of our knowledge and from our study of the related works presented in Section 2 and the academic tools, no work implements all the framework components to face the challenges in a Big Data architecture. Industrials attempting to be GDPR compliant and to avoid GDPR penalties reinforced their products by adding GDPR features or providing GDPR tools for that purpose.

6.2. Industrial GDPR tools

In addition to academic tools, industrial security tools have been proposed and classified in our comparative table. Generally, companies providing these tools do not provide details on the implementation. We identify some of them in this section:

- **The Absolute Platform:** This tool provides visibility and control. It addresses GDPR prerequisites by observing and verifying PII (Personally identifiable information), avoiding data breaches and automating remediation. The main features of this tool are not very well detailed but, we can consider that it partially provides the Collection manager component functionalities [1].
- **AlgoSec:** AlgoSec is an automation solution for network security policy management. With AlgoSec you can accurately process security policy changes in minutes or hours, not days or weeks. Using intelligent, highly customizable workflows AlgoSec streamlines and automates the entire security policy change process from planning and design to proactive risk analysis, implementation on the device, validation and auditing. This tool partially addresses Access and Consent manager [3].

- **Collibra:** The Collibra Platform is built on a foundation of data control and governance to ensure the security of user and enterprise data. It creates and maintains a rigorous control security framework built around regulatory, legal and statutory requirements as well as industry best practices. This tool addresses data privacy rights and finding data by cataloging and lineage techniques to get the full story behind data [26].

- **Compliance forge:** It offers project management tools for privacy by design. It uses automation to integrate security and privacy controls into standard project management processes [27].

- **MY DATA manager:** MY DATA manager was developed to achieve GDPR challenges by simplifying and automating the management of GDPR compliance specifications and processes. It provides a set of features such as data mapping, compliance assessment, data inventory activities and data explorer [48].

- **Alien Vault USM:** This tool helps to detect data breaches and monitor data security. The unified platform centralizes essential capabilities like asset discovery, vulnerability scanning, intrusion detection, behavioral monitoring, log management and threat intelligence updates [78].

- **BigId:** This tool assures data minimisation through duplication discovery and correlation. It satisfies customer data portability, supports and enables right-to-be-forgotten. In addition, it reveals enforcement of customer consent for personal data collection, data residency flows and risk profiling with breach notification windows [17].

- **BWise GDPR Compliance solution:** This tool helps to build data views, data control and compliance. It helps to efficiently collect, access, transfer or share data assets and safeguard data privacy and data protection [22].

- **Consentua:** This is a consent choice and control tool that enables users to choose and control their personal data. It empowers an increasingly trusted and straightforward relationship between the client and the service provider. It captures consent throughout the customer journey as needed. Then it gives the user the ability to control data processing in real-time. Finally it gives a picture to know how, why and where consent was collected [28].

- **PrivacyPerfect:** This work is composed of a set of tools such as assessment, processing and dashboard tools specially designed for chief privacy officers, reports, legal processing grounds and graphical overviews [56].

- **Hashicorp Vault:** Hashicorp Vault manages secrets and protects sensitive data securely, stores and tightly controls access to tokens, passwords, certificates, encryption keys for protecting secrets and other sensitive data using a UI, CLI, or HTTP API. This product addresses a set of IT design requirements such as establishing control with policies and rules, accessing controls and protecting the Data. Also, it is considered by the Hashicorp company as a product for GDPR compliance [42].

- **One Trust:** It automates the intake and fulfillment of consumer and subject rights requests. Also, it leverages intelligent risk mitigation to discover and address risks faster. It provides assessment automation and targeted data discovery. It addresses a set of GDPR articles in order to provide compliance. The main features are: assessment automation, data inventory and mapping and targeted data discovery [49].

- **Skyhigh Networks:** This helps gain complete visibility into data, context, and user behavior across all cloud services and devices.[71].

6.3. Apache tools

Apache has developed a set of tools to provide security features to Big data systems architectures. These technologies can be used to address parts of GDPR requirements when used in a good way. Here are some popular solutions:

- **Apache Eagle:** Apache Eagle is an open-source solution for identifying security and performance issues instantly on Big Data platforms like Apache Hadoop and Apache Spark [12]. It analyzes data activities and daemon logs. It provides a state of the art alert engine to identify a security breach, performance issues and shows insights [7].

- **Apache Atlas:** Apache Atlas is an open-source solution used for data tagging. It provides open metadata management and governance capacities for organizations to make a catalog of their data assets, classify and govern these assets and to provide collaboration capabilities around these data assets for data scientists, analysts and the data governance team. It helps implementing the Finding Data and classification IT design requirements in order to build a data flow graph to track data [6].

- **Apache Ranger:** Apache Ranger is an open solution that helps developers to enable, monitor and manage the entire data security across the Hadoop platform. The vision with Ranger is to provide a framework for the central administration of security policies and user access monitoring [11].

6.4. A comparative study

The comparison of the studied tools is tedious work since they come from different communities and target different objectives and variant application contexts. We propose situating these tools against the framework components to measure their compliance with GDPR and also for reuse purposes. Indeed, instead of reinventing the wheel, some ideas and implementations can be reused for designed Big Data systems, even though the work context can be different. Consequently, a global picture of these solutions and their implemented parts of the regulation can be very helpful to Big Data system designers. Using the defined framework, we can set up the comparative table below. We adopt the following notations:

- ✓: indicates that the component is implemented by the referenced work.

- ∼: indicates that the component is partially addressed by the referenced work and some of its IT design requirements and features are not implemented.

- ×: indicates that the component is not implemented by the referenced work.

In Table 3, the framework components defined in Figure 3 are partially implemented in each solution. There are no tools that implement all components. More precisely, we can see that the components that are most covered are C1, C2 and C8. They mainly focus on security policy definition and management. This can be explained by the fact that involving people in the definition of their security constraints and the tracking of their data is the focus of many research works even before GDPR was adopted. Indeed, providing practical and intuitive API for users, not necessarily experts in security, was considered, for many years, a priority in privacy-sensitive systems like e-health and other IoT systems. On the other hand, the C3 component addressing data annotation is less implemented compared to the other components. With data heterogeneity and the multi-sources of Big data, data annotation becomes necessary for tracking and controlling data flows. This technique is rarely required in small systems with a uniform data format and source. Also, C9 and C10 are considered only in few recent works. It is actually mewly created requirement with GDPR which consists of interfacing with the data protection officer PDO and the supervisory authority.

7. The Framework implementation and application

In this section, we propose an implementation of the framework based on the tools studied in Section 6. We selected from Table 3 the best candidate technology chosen for each component in the context of our use case. Then, we present the framework application in order to ameliorate a previous work on GDPR-compliance in e-health systems. We describe the framework usage and evaluate its overhead on the application performance.

Our implementation is based on Apache Ranger [11] and Atlas [6]. Indeed, these technologies represent the defacto standard as a governance layer in Big Data Systems. Furthermore, following Table 3, Apache Ranger is already adopted for masking features in C4, Access control in C6, C5 and C7 and auditing information in C9 and C10. Also, Apache Atlas provides a global view for the collected data as data lineage in C3, C5 and C6 (data discovery) and allows data annotation. Much more, it is configurable with the chosen Apache Ranger to translate the defined tags or annotation automatically to policies (Tag-based-policy). The main functionalities of these standards are as follows:

GDPR Tools by Category		Our framework Components									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
GDPR analysis	GDPR for healthcare [79]	×	×	×	×	×	×	×	×	×	×
	GDPR in Health Clinics[46]	×	×	×	×	×	×	×	×	×	×
	GDPR investigation [41]	×	×	~	~	×	~	×	×	×	×
Academic GDPR tools	PrivacyTracker [39]	~	×	✓	~	×	×	✓	×	×	×
	Storage system for GDPR [70]	×	×	✓	✓	×	✓	×	~	×	×
	IoT Databox [29]	~	✓	~	×	×	~	✓	~	×	×
	GDPR Controller [61]	✓	✓	×	✓	×	×	×	~	~	×
	TagUBig [36]	✓	~	×	✓	~	×	×	×	~	×
	Consent management [23]	✓	✓	~	×	×	×	×	✓	×	×
	Special privacy [18]	✓	✓	~	~	×	~	×	✓	✓	~
	GDPR in smart home systems [4]	~	✓	~	×	✓	~	~	✓	×	×
	SAC [73]	×	×	~	×	~	~	×	×	~	~
Industrial GDPR tools	The Absolute Platform [1]	~	×	✓	~	×	~	×	~	~	~
	Alien Vault USM [78]	✓	✓	~	×	~	×	~	~	✓	~
	BigId [17]	✓	✓	~	×	×	×	×	✓	×	×
	BWise GDPR solution [22]	✓	×	✓	×	×	×	✓	~	~	~
	Consentua [28]	~	✓	~	~	×	×	×	✓	×	×
	PrivacyPerfect [56]	✓	×	~	×	~	×	×	×	~	×
	Algosec [3]	~	✓	~	×	~	✓	✓	~	~	~
	Hashicorp Vault [42]	×	×	~	✓	~	✓	×	~	~	~
	Collibra [26]	✓	~	✓	×	~	×	~	×	~	×
	One Trust [49]	✓	✓	✓	~	~	~	~	~	✓	~
	Skyhigh Networks [71]	×	✓	~	×	~	×	×	~	×	~
	Compliance forge [27]	~	✓	~	×	~	×	×	✓	~	×
	MY DATA manager [48]	~	✓	~	×	~	×	×	~	✓	~
Apache tools	Apache Eagle [7]	×	×	~	~	~	~	~	✓	×	×
	Apache Atlas [6]	✓	~	✓	×	✓	✓	×	~	✓	×
	Apache Ranger [11]	~	×	~	✓	×	✓	✓	✓	~	×

Table 3
Comparative Table of the IT GDPR tools

- The Atlas-Ranger integration unites the data classification and metadata store capabilities of Atlas with security enforcement in Ranger. Ranger implements dynamic classification-based security policies (Tag-based-policy). Ranger's centralized platform empowers data administrators to define security policy based on Atlas metadata tags or attributes and apply this policy in real-time to the entire hierarchy of entities including databases, tables, and columns, thereby preventing security violations. Ranger tags are attribute-based, every tag can have attributes. Tag attribute values are used in the tag-based policies to control the authorization decision. When configuring Atlas-Ranger to work together, TagSync is activated. Ranger TagSync is used to synchronize the tag store with the external metadata service Apache Atlas. It receives tag details from Apache Atlas via change notifications. As tags are added to updated or deleted from resources in Apache Atlas, Ranger TagSync receives notifications and updates the tag store in order to keep the policy implementation valid.

- For certain business use cases, data should have an expiration date for business usage (Data Time To Live). This case can be achieved with Atlas and Ranger. Apache Atlas can assign expiration dates to a data tag. Ranger inherits the expiration date and automatically denies access to the tagged data after the expiration date. This feature helps a lot in controlling data storage and data access in a GDPR context.

- Controlling the data access with Location-specific access policies similar to time-based access policies. In help taking consideration of the geographical location of the user when accessing the data.

- Visualizing data lineage in the Big Data application delivering a complete view of data movement across several analytic engines such as Apache Storm, Kafka, Falcon, Hive and recently Spark. As this tracking is carried out at the platform level, any application that uses these engines will be natively tracked with Atlas and secured with Ranger.

Apache Ranger and Apache Atlas represent the core of our framework implementation. We take the advantage of data interception techniques, the connectors to the manager layer and all the described features of data lineage construction, policy checking and enforcement. In our implementation, we added the following functionalities:

- A policy model to express the DS consent following an extended GDPR *5W* model.
- A security policy configuration checker compliant with the *5W* model.
- Attribute-based Encryption using CP-ABE.
- A set of user-friendly interfaces for the DS and DPO.
- A notification system about security breaches using Kafka.

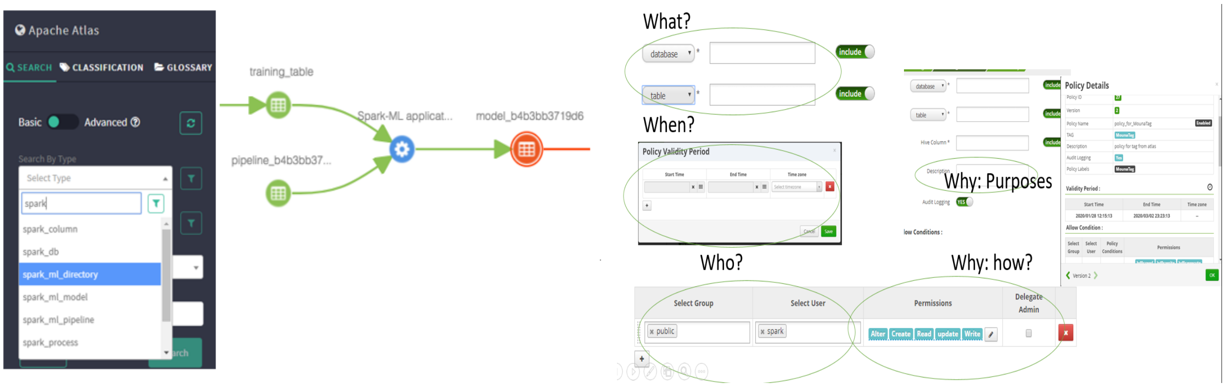
Regarding the security policy model, we proposed a taxonomy for privacy policies called *5W* [4] but other taxonomies can be used. In the *5W* model, the DS is asked to respond to *5W* questions: what data to be processed? why? how and where his data are stored? who can access it and why? is it up-to-date and accurate? and how long will he keep it for? Following the defined *5W* policy, a configuration checking is performed. For this security policy checking, we proposed in a previous work the following formal checking [4]: Let D be the data sets D = {d1, d2, .., dn } and P the set of processors P = {p1, p2, .., pn }, E the set of storage spaces E = {e1, e2, .., in } and L the set of labels (DS preferences) in the form of *5W* GDPR policy. For an incoming data d which is annotated with the *5W* policy (with the following tags *what*, *who*, *when*, *where*, *why*), the security configuration is correct if it meets user preference L regarding the same data owners. More formally, the configuration is accepted if we have for a function S such that S: P $\cup E \cup D \rightarrow$ L:

- If we have p in P, d in D. A process p is authorized to process a data d if: L (d) $\subseteq L(p)\,:\, d.what == p.what, d.why == p.why, d.where == p.where, d.who \subseteq p.who, d.when > p.when.$
- If we have e in E, d in D. A storage space e is authorized to store a data d if: L (d) $\subseteq L(e)\,:\, d.what == e.what, d.why == e.why, d.where == e.where, d.who \subseteq e.who, d.when > e.when.$

After installing and configuring Atlas-Ranger, we implemented in Atlas a GDPR classification following the GDPR *5W* policy [4]. We provide a user interface so that a DS can define his constraints following the *5W* policy. Then, it is sent to Atlas via a REST API to tag the collected DS data. Atlas annotates the adequate selected data captured by its hooks with *5W* tags and propagates it to the rest of the data lineage to control the calculated data.

The policy configuration checker is executed before enforcing this data. We capture the new annotated lineage from Atlas via its REST API, we execute the checker to verify if a problem is detected (we used a java implementation for the policy configuration checker). The policy checker verifies that each data annotation takes into account the preferences of the DS and that each processing or storage layer has access to authorized data only. This task is performed each time new data are received or whenever newly calculated data are obtained as explained in the proposed security model [4]. Once the configuration is performed, if a problem exists a notification is sent to the DS for consent update. If no problem is detected the *5W* policy is enforced and data are secured. One of the *5W* policy attributes is if data should be encrypted or not, if yes a CP-ABE encryption algorithm is executed [61].

We have Atlas-Ranger are configured, then the TagSync module is activated. More precisely, policies are stored and enforced in Apache Ranger as tag-based policies using the Tag sync module automatically. TagSync is used to populate the tag store from the tag details available in an external system in our case Apache Atlas [6]. TagSync is a daemon process. In the currently used release, ranger-TagSync supports receiving tag details from Apache Atlas via change notifications. As tags are added/updated/deleted to resources in Apache Atlas, ranger-TagSync would receive notifications and update the tag store. The GDPR *5W* policies are enforced as presented in Figure 4(b) as "Tag-Based Policy". Each W is automatically mapped to one of the Apache Ranger policies, as illustrated in Figure 4(b). The topic of the *What* tag can be a database, a table or a column and the masking option will be a simple Tag as PII (for the value "true" in the policy) if the data are Tagged like PII, they will be hidden by Apache Ranger (masking data). The *When* tag is a period for the DTTL, we can set the start time and the end time. The collection time will be compared to the



(a) Data flow using Apache Atlas for the Spark process

(b) Implementation of a *5W* security policies in Apache Ranger

Figure 4: Apache Atlas and Apache Ranger configuration

current time (system date) for each request. The *Who* tag contains two options: the owner is the connected user and the owner of the Token, the processors are mentioned and grouped by user group. The *Why* tag also contains two options: the purpose is defined in the description of the policy (in future work, we will work on the semantics of the purpose as defined by the DS) and the *how* option is defined as permissions (alter, create, read, update, write). Finally, the *Where* tag is detailed in three tags: the source is verified in the security checking process as the first data collection point, the destination is all the storage spaces in the graph. The transfer right is represented as a boolean type and controlled by the Location-specific access feature of Ranger.

In addition to the provided consent form, we displayed a user-friendly data flow graph for the DS from the JSON format captured from Apache Atlas lineage using his REST API. Then, we redefined the data flow graph using a graph calculation framework: Apache TinkerPop [13] with a light representation compared to Apache Atlas lineage in figure 4(a). Also, Apache Atlas does not provide a Hook (connector) to collect metadata from the Apache Spark process. This is why we used a recent implementation of Spark-Atlas-Connector SAC [74] from our Table 3 to obtain data lineage of the data processed by Apache Spark. After performing the right configuration, Apache Atlas can detect all spark processes as illustrated in Figure 4(a). Here, our framework is ready for any processing type even for complex Spark process.
Figure 5 displays the main technical components selected for the framework implementation.

- **C1 implementation:** For task orchestration and inter-component communication, Apache Kafka [10] is classically used [61]. Indeed, the publish/subscribe broker of Kafka allows for loosely coupled and scalable communication between the different framework components. Components are notified about new data, new security policies and security breaches when they occur. All components are information producers and also information consumers which allow C1 to play the role of a central hub in the management layer. A particular consumer component is the policy checker that relies on the annotated graph for checking policy compliance between annotated data from one side and storage spaces and processes from another side [61].

- **C2 implementation:** We provide a user interface so that a DS can define his constraints following the *5W* policy and track his data in all the Big Data life-cycle with fine granularity.

- **C3 implementation:** In the Gateway we need to filter the collected data in order to provide data minimisation. The DS consent is used as annotation for data and access control policy at the same time in Atlas-Ranger.

- **C4 implementation:** We adopt the same implementation of Crypto-Engine, using a particular algorithm of ABE called CP-ABE [61]. Attribute-Based Encryption (ABE) is a form of public-key encryption. The ABE algorithms represent a good candidate to achieve privacy and fine-grained access control for Big Data applications running on Cloud servers. Furthermore, in [45] shows that the proposed scheme can not only achieve fine-grained access control but also support resisting the collusive attack. Our choice is additionally motivated by CP-ABE evaluation in [61] where authors show an acceptable overhead. We confirm this result in the evaluation part of our work.

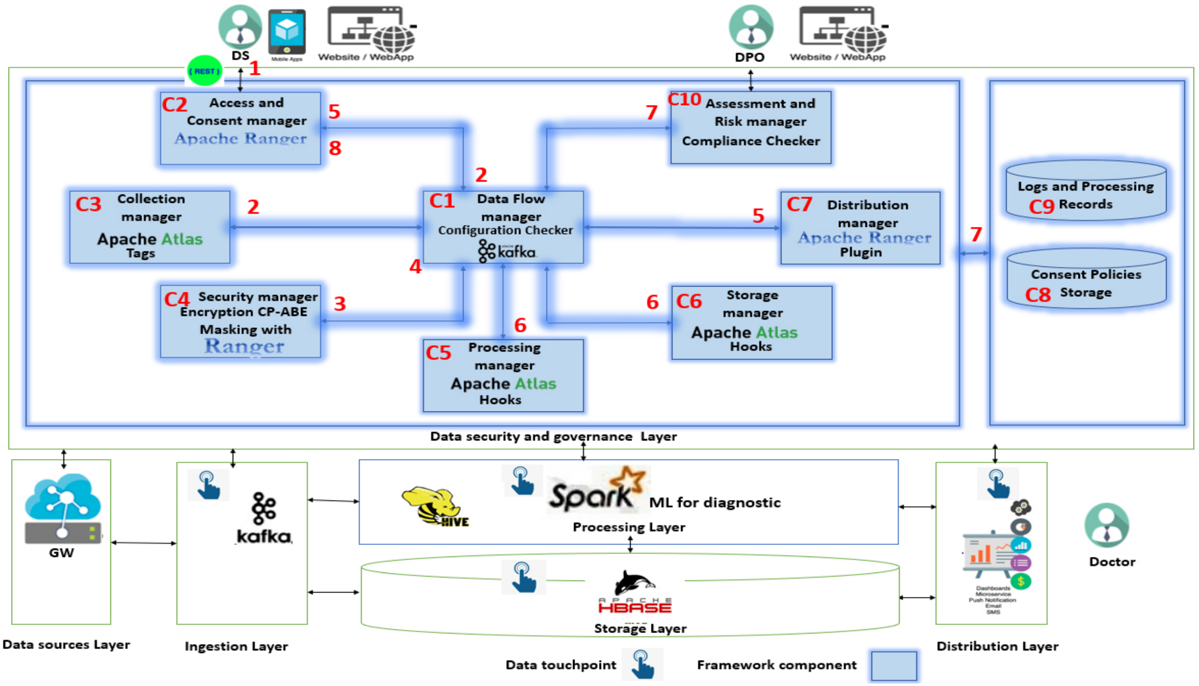

DS
Mobile Apps
Website / WebApp
DPO
Website / WebApp
REST
1
C2 Access and Consent manager Apache Ranger
5
8
7
C10 Assessment and Risk manager Compliance Checker
2
C3 Collection manager Apache Atlas Tags
2
C1 Data Flow manager Configuration Checker kafka
5
C7 Distribution manager Apache Ranger Plugin
7
4
C4 Security manager Encryption CP-ABE Masking with Ranger
3
6
6
C6 Storage manager Apache Atlas Hooks
C5 Processing manager Apache Atlas Hooks
Logs and Processing C9 Records
Consent Policies C8 Storage
Data security and governance Layer
GW
kafka
HIVE
Spark
ML for diagnostic
Processing Layer
HBASE
Storage Layer
Dashboards Microservice Push Notification Email SMS
Doctor
Data sources Layer
Ingestion Layer
Distribution Layer
Data touchpoint
Framework component

Figure 5: The framework implementation

- **C5 and C6 implementation:** These two components are responsible for controlling the storage and processing of data. Thanks to Apache Atlas and Apache Ranger, we collect in real-time the process scope and manage the stored data, the storage location and the "data time to live" against the consent policy defined by the DS. Every processing and storage process is stored in Logs and Processing Records C9 database via a specified Kafka topic of C1 in addition to logs provided by Apache Ranger. In case of a breach, notification is sent to the DS and to the DPO to notify them. We adopted Apache Atlas hooks. Apache Atlas Hooks (Atlas connectors) are provided in order to collect all meta-data for transparency and accountability purposes and Ranger plugin to verify if storage space or a process is allowed to have Data activating also the policy configuration checker. Because in Big Data application, we need complex processing such as ML process (Machine Learning), we used Apache Spark in batch mode to process the stored collected data.

- **C7 implementation:** The distribution layer helps to control users' or applications' touchpoints with the newly calculated or stored data by evaluating access control using a Ranger plugin to evaluate the policy retrieved from C8. The data query is captured by the distribution manager and then sent to the Ranger plugin for evaluation and access control. While authorizing an access request, Apache Ranger plugin evaluates applicable Ranger policies for the resource being accessed.

- **C10 implementation:** If we see Table 3, Assessment and Risk management are not really implemented in existent systems except in [18]. In our work, we use logs and processing records to feed the Assessment and Risk manager component, C10, in order to demonstrate compliance for the Data Protection Officer. C10 provides a security dashboard and real-time logs. This component uses the consent policies together with processing scopes as business policies (the processing logs provided by Kafka) to check that data processing and sharing comply with the relevant usage control policies. The Logs and Processing Records database is alimented with ranger logs and collected from Ranger plugin in the distribution manager where the access of all services is controlled.

The output of the framework implementation is detailed and illustrated in the next section as a real e-health application with real data set.

7.1. Application to e-health GDPR-compliance

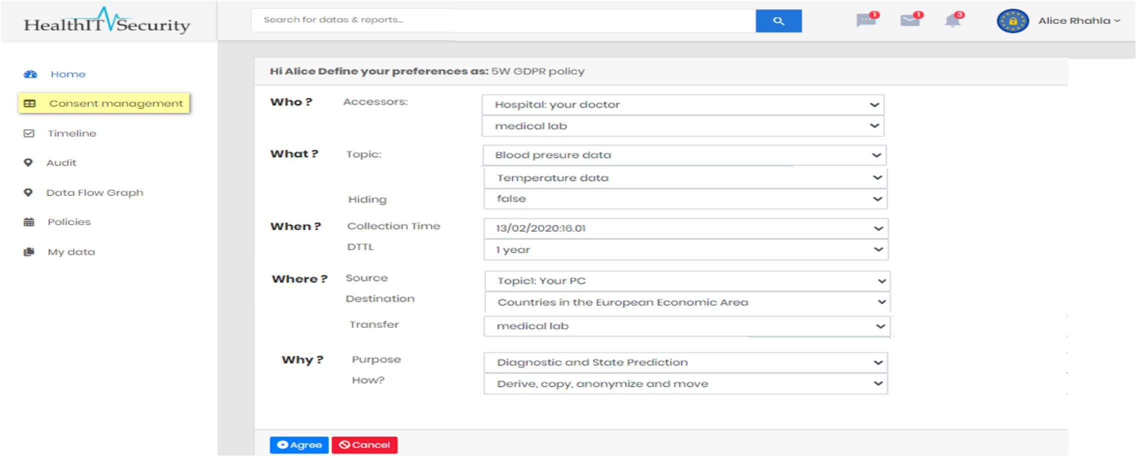
In this section, we used our implemented framework components to improve our previous work [61].This work was developed as part of a client project for a digital healthcare publisher that offers software for remote monitoring, diagnostic assistance, and digital therapeutic education authorizing remote patient care. Their products assist medical staff in taking care of their patients remotely. They are designed around four pillars: Remote monitoring (make it possible to monitor patients remotely and better adapt their care.), diagnostic and prediction (assist caregivers in the detection of relapses and therapeutic toxicities.), Support, administrative, and health statistic (allow the Therapeutic Education of Patients from a distance to guide and reassure them and prepare recommendation based on historical data). Their applications and services are IoT based. So, we consider a classical e-health IoT system with a set of sensors collecting patients' state (Let us call Alice the DS in our use case) and sending this data to a Gateway. A set of services collaborate to analyze received data and to calculate diagnosis (Processing and Storage Layer). Typically, Alice's data are collected to develop diagnosis (diagnostic service) and predict changes in Alice's state (state prediction service). Data are selectively accessible by different services depending on the type of disease and by research laboratories according to Alice's consent (Distribution Layer), for example, if Alice data is communicated to a research lab or other services without Alice consent, this event has to be detected and data cannot be communicated without Alice consent. Shared and critical data must be protected against unauthorized access while providing accurate and fine-grain access control for the authorized actors. This use case highlights the need for the setting of multilevel and dynamic security policies. Alice can delegate her data control to third-parties like the medical staff or her doctor Bob or a parent. Alice as a Data subject (DS) has the right to define and modify her security policies (her consent) and is notified if they are violated (used by auxiliary services or administrative services). In our previous work, A multi-level security model is proposed to describe fine-grained access control policies. Data subjects can track their data flows and can be notified about any illicit access. Data are annotated following the security constraints and access control is executed at run time. However, it was not clear to us what is exactly covered by GDPR and what is missing. The framework allowed us to accomplish this task.

As shown in the comparative Table 3, many functionalities are missing in our previous work [61]. More precisely, three components are implemented (C1, C2 and C4), two components are partially implemented (C8 and C9) and five components (C3, C5, C6, C7 and C10) are missing. In the new version of this work, we reuse the implemented framework for GDPR-compliance with a python/spark process-based pipeline. In the use case implementation, we used a batch mode processing: data coming from sensors or GW are captured by the adequate Kafka topic in the ingestion layer, then stored in the hospital storage layer (Hbase). The stored data will be processed by Apache Spark for diagnostic purpose and hive for statistical and administrative purpose. Finally, the result is displayed to Alice's doctor in the services layer (It is a typical Big Data pipeline).

In the use case architecture of Figure 5, we highlight the different data touchpoints that need to be controlled by the proposed framework: the GW touchpoint, the ingestion layer touchpoint, the storage layer touchpoint, the processing layer touchpoint and finally the distribution layer touchpoint.

Let us consider the following scenario for the implemented use case and check that the DS consent is respected. We suppose that we have 4 services in our use case: diagnostic service, state prediction service, statistical and administrative service and auxiliary service. The framework is used to check GDPR compliance of these services and if Alice's consent is respected. A user interface asks Alice to respond to a *5W* form: she must provide what data to be processed? why? how and where her data are stored? who can access it and why? is it up-to-date and accurate? and how long will she keep it for?. Alice's policy says that the following IoT data can be collected: the blood pressure, heart rate, temperature and location. These data are stored in the hospital's servers in the EU. The hospital (which plays the role of a controller) additionally asks if these data can be shared and used by the medical lab. Alice accepts this option of data forwarding. Figure 6 shows the consent request presented as a *5W* GDPR form to be felt by Alice.

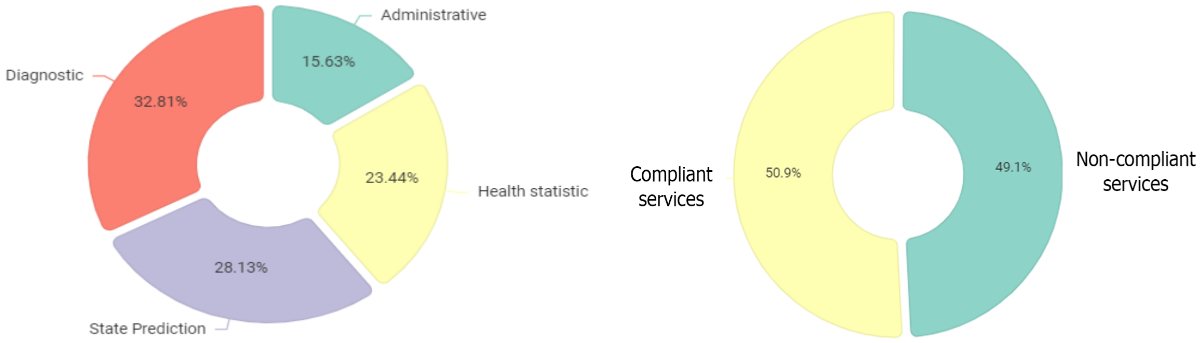
After collecting and securing her data, the controller as we mentioned need to demonstrate that Alice's consent is implemented as desired and designed to both Alice as a DS and to the DPO for transparency and compliance purposes. To demonstrate compliance to the DPO an Assessment and Risk manager dashboard is provided: Here a compliance checking is performed as explained in the components implementation. Figure 7 shows some result of the compliance dashboard that can be used by a controller to demonstrate accountability to the DPO: First an investigation is presented to the DPO, here for each data type, we can see all the processing, purposes, storage spaces related to. If we take the



Hi Alice Define your preferences as: 5W GDPR policy

Who ?	Accessors:	Hospital: your doctor
		medical lab
What ?	Topic:	Blood presure data
		Temperature data
	Hiding	false
When ?	Collection Time	13/02/2020:16.01
	DTTL	1 year
Where ?	Source	Topic1: Your PC
	Destination	Countries in the European Economic Area
	Transfer	medical lab
Why ?	Purpose	Diagnostic and State Prediction
	How?	Derive, copy, anonymize and move

Figure 6: A GDPR 5W Consent: Alice's consent definition responding to the 5W questions



(a) DPO Investigation process: purposes related to blood pressure data

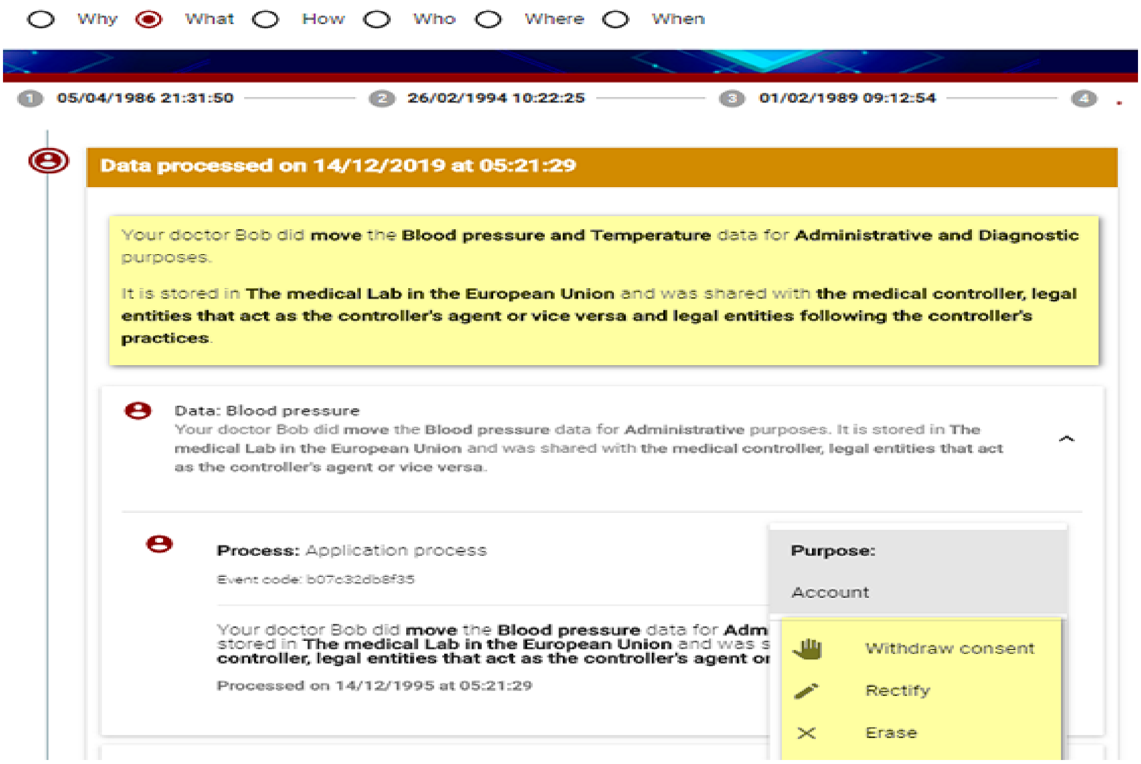
(b) Compliance checking: Was consent given?

Figure 7: DPO investigation and compliance checking

example of blood pressure data, Figure 7(a) shows the different services related to the processing of this data: diagnostic, state prediction, administrative and auxiliary services. Here, the investigation process shows that Alice's medical data are used by the 4 services. Following Alice's consent, blood pressure data is not collected for administrative purposes or auxiliary purposes. The compliance checking is automatically starting the comparison after building the *5W* business policy as a result of the investigation process. In Figure 7(b), compliant and non-compliant services are detected in the hospital as a GDPR status. In case of compliance failure, it is important to explain which parts of the business policies are not adequate and cause compliance checking to fail. It may also be useful to suggest possible corrections and generate a report to the SA.

Finally, a Transparency and Compliance dashboard is also provided to Alice, as a DS, to track and control her data. First, a timeline is illustrated using the *5W* consent attributes as a filter for all details on the processing of her data (Why, What, How, Who, Where and When) in order to check that her consent is respected. For example in Figure 8, Alice wants to know all information about the processing of her blood pressure data. She performs a filter by "What" annotation. She gets in her timeline, when the process of her blood pressure takes place (date and hour), who collected her blood pressure data, for what purpose and how it is used.

Alice can now decide to revoke the given consent and ask the hospital to delete all of her data as highlighted in Figure 8. In addition to her timeline, Alice needs to see her data flow graph and how her consent is propagated. Therefore, the controller provides a simple representation: a high-level data flow to display to Alice in her user interface with fine granularity by extracting exactly the data used in the query or the process. As a result, we have obtained a user-friendly data flow graph for our use case, as shown in Figure 9. We have 3 types of nodes: data D, process P or storage space S



○ Why ● What ○ How ○ Who ○ Where ○ When

1 05/04/1986 21:31:50 — 2 26/02/1994 10:22:25 — 3 01/02/1989 09:12:54 — 4

Data processed on 14/12/2019 at 05:21:29

Your doctor Bob did **move** the **Blood pressure and Temperature** data for **Administrative and Diagnostic** purposes.

It is stored in **The medical Lab in the European Union** and was shared with **the medical controller, legal entities that act as the controller's agent or vice versa and legal entities following the controller's practices**.

Data: Blood pressure

Your doctor Bob did **move** the **Blood pressure** data for **Administrative** purposes. It is stored in **The medical Lab in the European Union** and was shared with **the medical controller, legal entities that act as the controller's agent or vice versa**.

Process: Application process

Event code: b07c32db8f35

Your doctor Bob did **move** the **Blood pressure** data for **Adm**
stored in **The medical Lab in the European Union** and was s
controller, legal entities that act as the controller's agent o

Processed on 14/12/1995 at 05:21:29

Purpose:

Account

Withdraw consent

Rectify

Erase

Figure 8: Alice's Timeline for Consent tracking: Is consent implemented as designed

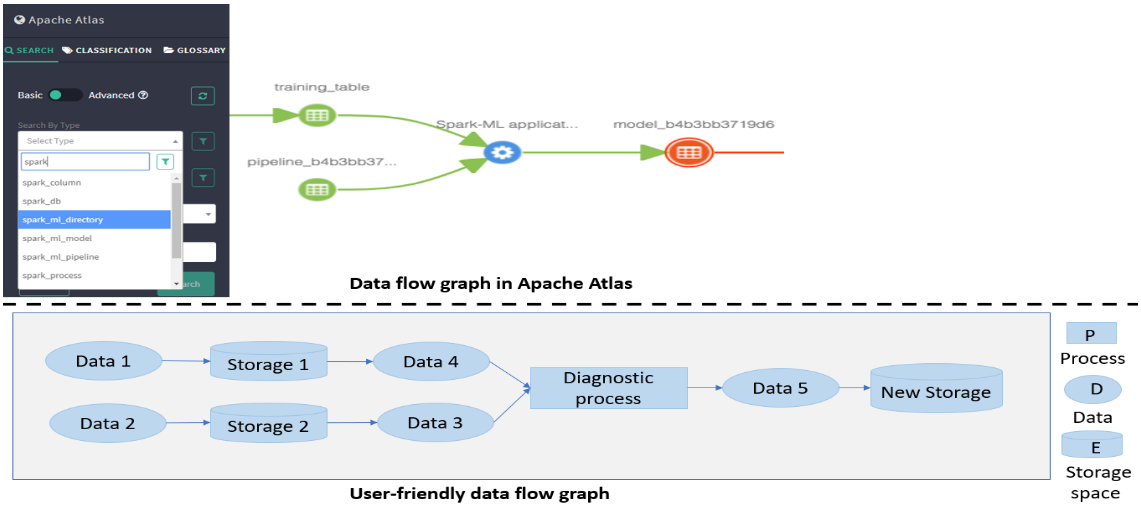


Figure 9: User-friendly data flow graph for Data tracking

as shown in Figure 9 and as used in the policy configuration checker model definition. As we see, Alice has a complete overview and full control of her collected data.

7.2. Evaluation of GDPR-compliance overhead

We distinguish two evaluations: performance overhead and engineering overhead. In the first evaluation, we measure the execution time for security checking and encryption/decryption tasks. We also compare our use-case execution with the introduced security layer (implementing our framework) and without. For the engineering overhead, we describe the engineering effort that is required to embed the security management layer to an existent Big Data solution. We explain that, since our implementation is based on Big Data technology (Apache Kafka, Apache Atlas, Apache Ranger), the performance and engineering overheads are minimal.

7.2.1. Performance overhead

Our evaluation settings are as follows:

- **Environment:** The evaluation is carried out on a PC as a gateway, with Intel Core i5 up to 2.4 GHz and 8 GB of memory. The use case application is implemented in a Hadoop ecosystem on a cluster with 300GB of memory and 24 cores. The cluster has three workers, each one with 8 cores and 100GB.
- **Data:** In this implementation, we used a real data set extracted from a Proxym-IT client database: an IoT e-health platform (e-Health, Big Data, Quantified Self, and Digital Health). Our industrial client maintains the patients' data which doubles in size every 11 months. The health services and products have supported more than 500,000 patients since 2013. Patients' data are collected from different IoT devices such as smart band, smart bathroom scales, smart blood pressure monitors, etc. The platform enables the creation of intelligent, rigorous, and engaging digital therapeutic supports and analyzes the collected data. In this evaluation, we focused on blood pressure measures. We extracted these measures for 100,000 patients collected by smart blood pressure monitors. We set the smart blood pressure monitors to send one sensor-data/10 minutes so, 144 measures per day for each patient. The obtained data set contains almost 2 592 million measures of 100,000 patients for about 5 months. We used the provided data set, it contains 76 attributes, but our experiments refer to using a subset of them. In particular, the following attributes are used: Patient ID(An id number representing the patient), Patient Name (A text string representing the patient name), Origin(A text string representing the sources of data collected: device ID) Age (A text string representing the age of the patient), Gender (A text string representing the gender of the patient), height (cm), weight(kg), apHi(Systolic blood pressure), ap_lo(Diastolic blood pressure), cholesterol(1: normal, 2: above normal, 3: well above normal), gluc(1: normal, 2: above normal, 3: well above normal), smoke(A text string indicates whether the patient smokes or not), temp(a text string indicates the patient's temperature degree).
- **Type of processing:** In our evaluation, we evaluated one type of service performing simple operations. In the state prediction and monitoring service, medical staff is performing patients state monitoring based on their historical data in batch mode to better adapt their care and diagnostics. On the other hand, Several patients have made many queries to send additional data or get data about them. In this service process, we are performing many python/spark operations such as: "grouping", "selecting", "joining", "sorting", "fitting", "predicting".

Considering the provided settings, we compare the execution time of two cases with and without the data security and governance layer for GDPR-compliance. We measure the response time starting from the ingestion layer to the services layer (the diagnostic result is displayed to the doctor). We vary the number of users using the application (patients/second) to reach up to 600 users/second.

The measured response time does not include the time for sending data from the GW to the ingestion point and the time for accessing data in the storage layer. We have 5 touch-points: in the GW data need to be encrypted, in the ingestion and storage layer data need to be decrypted for use, in the processing layer Spark process need to be controlled and finally, in the services layer access to data by the doctor need to be controlled. Controlling the spark process and the access to data refer to applying the attributes of the enforced *5W* policy by Ranger plugins and executing the policy configuration checker to keep policies implementation valid. We obtained the result shown in Figure 10. The graph shows that compared to the classical use case with no GDPR compliance, we still have an acceptable latency with low variations. This variation is due to the different data interceptions and processing at touchpoints: Process access control to data, policy configuration checker and encryption/decryption operations. To have more details about the impact of each interception, we evaluate each one separately.

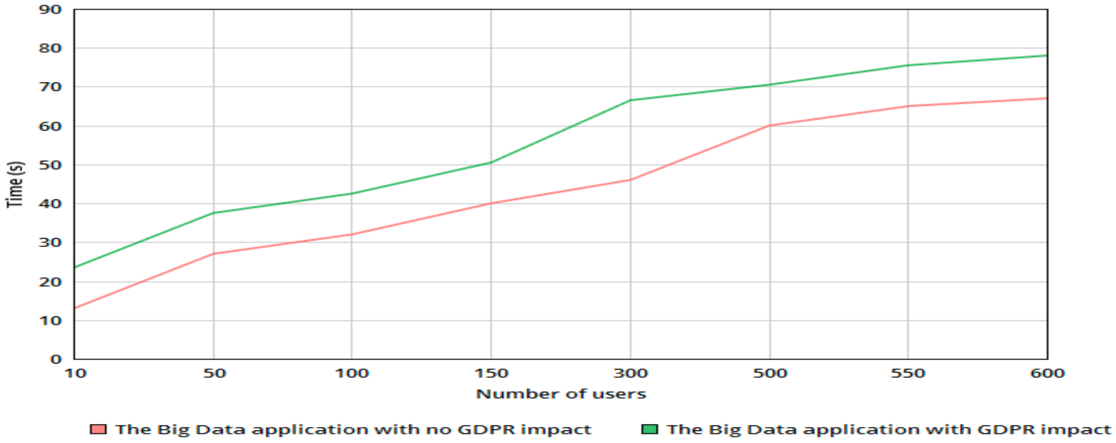
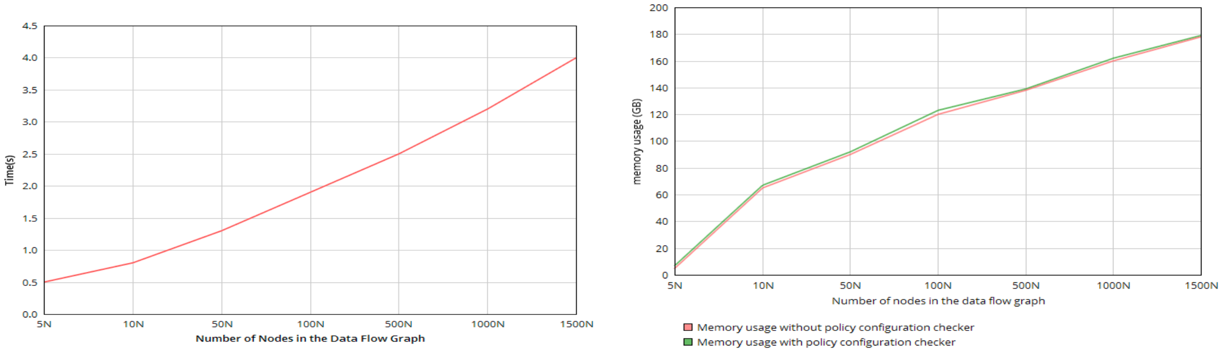


Figure 10: Global evaluation: the impact of the framework implementation on a classical e-health use case



(a) Policy configuration checker execution time

(b) Policy configuration checker memory consumption

Figure 11: Performance evaluation of Policy configuration checker

First, we evaluate the access control to data. Queries to data are captured by the Distribution manager to decide whether data can be communicated to services. The previous operation may introduce a huge processing delay if the access control verification is performed for many data items. However, thanks to Apache Ranger scalability [11, 25], this processing overhead is reduced to few milliseconds.
Second, we evaluate the performance of our Policy Configuration Checker that is, actually, an extension to the couple Atlas and Ranger and it may slow down their performance. This checker is executed at deployment time and run-time when the security configuration changes. The key measures to consider are memory and execution time. The memory depends on the number of edges E and the vertices V of the data flow graphs. The graph is generally composed of few edges and vertices but, in our evaluation, we are placed in complex Big Data applications composed of several data sources and several service destinations going through several processors. The worst case is reached when each destination is connected to all sources in the graph. In this assessment, we consider many nodes ranging from 5 to 1500 nodes and we consider fully connected graphs. Figure 11(a) shows the variation of the execution time for verifying policies in the considered data flow graphs. The execution time increases with the number of nodes in the data flow graph of Apache Atlas lineage. The execution time remains very acceptable, it does not exceed 4s even for scalable and fully interconnected graphs (1500 nodes).

For the memory consumption, we consider the same experimentation with nodes ranging from 5 to 1500 nodes

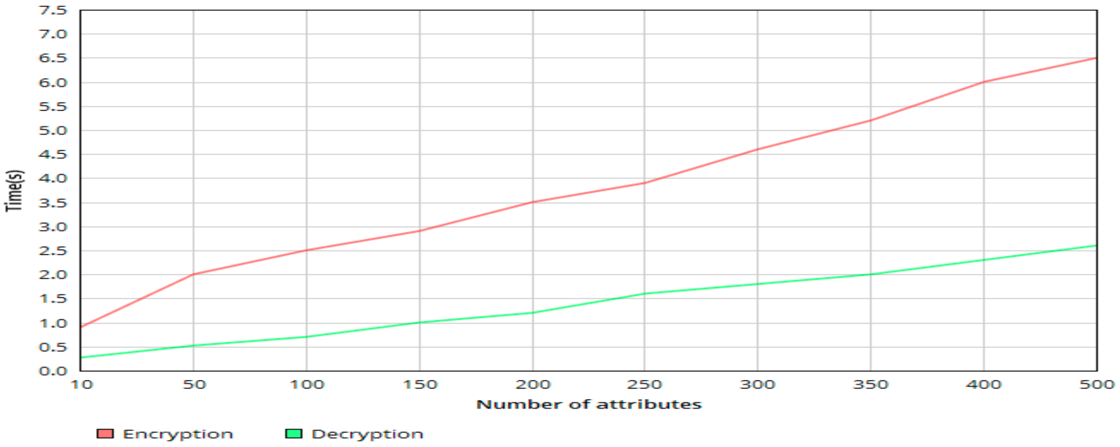


Figure 12: CP-ABE performance evaluation: Overhead of encryption/decryption algorithms

and we consider fully connected graphs. We compared the memory consumption with and without the framework integration. Figure 11(b) shows that the execution of policy configuration checker consumes slightly more memory compared to the initial evaluation. This confirms that memory depends on the number of edges E and the vertices V of the data flow graphs. In a Big Data context and from the shown low variation, we can mention that memory consumption is not affected by the execution of the policy configuration checker algorithms.

Finally, we evaluate the overhead of using CP-ABE for encryption. In addition to the evaluation presented in [61], we evaluate both the encryption and decryption time that increase linearly with the number of considered data attributes. We vary the number of attributes up to 500 and we consider 1GB of sensor data coming from IoT devices at the same time. Encryption or decryption operations are executed at the DS GW and at the ingestion layer. Even with this exaggerated configuration that is very unlikely to occur in IoT systems, the evaluation still provides acceptable execution times. Figure 12 shows that the time taken by the decryption and encryption algorithm varies linearly with the number of attributes and it is still acceptable compared to other evaluations [61].

7.2.2. Engineering overhead

In Big Data applications, the management layer with Apache Ranger and Atlas are classically installed and configured with their hooks for the processing and storage data touchpoints thanks to Apache Ambari [5] sandbox. We used Ambari for our framework installation and configuration comprising Big Data components (Apache Kafka, Apache Ranger, Apache Atlas, Apache Spark, etc). The hooks configuration depends on the technology used for each data touchpoint to push metadata changes to Apache Atlas for data control. For example, for the ingestion layer touchpoint, we need a Kafka hook to capture which data collected by Atlas, same for storage data touchpoint we need a hdfs/Hive hooks and for the processing a Spark hook. The developer has just to write his spark code on an installed notebook like Apache Zeppelin [14] on the provided sandbox with a connector as a hook configuration, or just run with a command line on the sandbox passing the connector as a parameter (bin/spark-shell –jars spark-atlas-connector_2.11-0.1.0-SNAPSHOT.jar). To summarize, since our implementation is based on Apache Ranger-Atlas solution, we use hooks and connector techniques provided by these technologies to embed the management layer. Similar hook approaches can be applied if other technologies are used for the framework implementation.

8. Conclusion and Future Work

This work aims at helping IT designers and developers understand GDPR and implement GDPR-compliant Big Data systems. For this, we analyze GDPR requirements and translate them to IT design requirements. Then, a framework is proposed that details the main components for GDPR compliance verification and implementation. To implement this framework, we classified and compared different tools related to GDPR implementation in Big Data systems. This comparison is guided by the identified IT requirements and the framework components designed in this paper.

An implementation is proposed extending the Big Data governance layer, Apache Ranger associated with Atlas. To validate this implementation part, we consider an e-health application, and we show how GDPR is respected using our solution without significant overhead on system performance or engineering effort.

The framework is the first step towards GDPR-compliance in Big Data systems. It allows simplifying GDPR understanding to the IT community and providing guidelines for Big Data GDPR compliance. Nevertheless, it is evaluated on a single kind of application using Kafka and spark technologies. We need to evaluate the framework on other kinds of Big Data pipelines. Furthermore, our use-case application is in batch mode. Real-time and streaming applications (using CEP [8] or Spark Stream) have more challenging requirements in terms of response time. It is important to check the GDPR-compliance overhead induced by our framework on real-time processing. Furthermore, the framework components' implementation can be enhanced by introducing other security implementations such as Lattice-Based encryption [31] and more user-friendly API for DS and DPO. As future work, we are interested in applying the framework on streaming applications and evaluating GDPR-compliance on this kind of application.

References

[1] Absolute, . The absolute platform. www.absolute.com. Online; accessed 01/15/2020.

[2] Akhigbe, O., Amyot, D., Richards, G., 2015. Information technology artifacts in the regulatory compliance of business processes: a meta-analysis, in: International Conference on E-Technologies, Springer. pp. 89–104.

[3] Algosec, . Algosec. https://www.algosec.com. Online; accessed 01/15/2020.

[4] Allegue, S., Rhahla, M., Abdellatif, T., 2019. Toward gdpr compliance in iot: Data controller integration in smart home systems. International Conference on Service Oriented Computing (ICSOC 2019): ISYCC workshop. ,Toulouse, France .

[5] Apache, . Apache ambari. https://ambari.apache.org/. Online; accessed 03/03/2020.

[6] Apache, a. Apache atlas. https://atlas.apache.org/. Online; accessed 01/15/2020.

[7] Apache, b. Apache eagle. https://eagle.apache.org/. Online; accessed 01/15/2020.

[8] Apache, a. Apache flink:complex event processing cep. https://ci.apache.org/projects/flink/flink-docs-stable/dev/libs/cep.html. Online; accessed 29/02/2020.

[9] Apache, b. Apache hadoop. https://hadoop.apache.org/. Online; accessed 03/02/2020.

[10] Apache, a. Apache kafka. https://kafka.apache.org/. Online; accessed 01/15/2020.

[11] Apache, b. Apache ranger. https://ranger.apache.org/. Online; accessed 01/15/2020.

[12] Apache, a. Apache spark. http://spark.apache.org/. Online; accessed 03/03/2020.

[13] Apache, b. Apache tinkerpop. http://tinkerpop.apache.org/. Online; accessed 03/03/2020.

[14] Apache, c. Apache zeppelin. https://zeppelin.apache.org/. Online; accessed 03/03/2020.

[15] Apache, d. Redis. https://redis.io/. Online; accessed 03/02/2020.

[16] Badii, C., Bellini, P., Difino, A., Nesi, P., 2020. Smart city iot platform respecting gdpr privacy and security aspects. IEEE Access 8, 23601–23623.

[17] BigId, . Bigid. https://bigid.com/eu-gdpr/. Online; accessed 01/15/2020.

[18] Bonatti, P.A., Kirrane, S., 2019. Big data and analytics in the age of the gdpr, in: 2019 IEEE International Congress on Big Data (BigDataCongress), IEEE. pp. 7–16.

[19] bpr4gdpr, . The business process re-engineering and functional toolkit for gdpr compliance project. https://www.bpr4gdpr.eu/. Online; accessed 15/01/2021.

[20] Brodin, M., 2019. A framework for gdpr compliance for small-and medium-sized enterprises. European Journal for Security Research 4, 243–264.

[21] Burt Kut, M., 2018. Key tension points and design guidelines for gdpr compliance: Designing for a news service application.

[22] BWise, . Bwise gdpr compliance solution. www.bwise.com/solutions/regulatory-compliance-management/global-data-protection-regulation-gdpr. Online; accessed 01/15/2020.

[23] Camilo, J., et al., 2019. Blockchain-based consent manager for gdpr compliance. Open Identity Summit 2019 .

[24] de Carvalho, R.M., Del Prete, C., Martin, Y.S., Rivero, R.M.A., Önen, M., Schiavo, F.P., Rumín, Á.C., Mouratidis, H., Yelmo, J.C., Koukovini, M.N., 2020. Protecting citizens' personal data and privacy: Joint effort from gdpr eu cluster research projects. SN Computer Science 1, 1–16.

[25] Cloudera, . Providing authorization with apache ranger. https://docs.cloudera.com/HDPDocuments/HDP3/HDP-3.1.4/authorization-ranger/sec_authorization_ranger.pdf. Online; accessed 03/06/2020.

[26] Collibra, 2019. Collibra. https://www.collibra.com/. Online; accessed 03/02/2020.

[27] Compliance forge, . compliance forge. https://www.complianceforge.com/product/security-by-design-privacy-by-design/. Online; accessed 03/02/2020.

[28] Consentua, . Consentua. https://consentua.com. Online; accessed 01/15/2020.

[29] Crabtree, A., Lodge, T., Colley, J., Greenhalgh, C., Glover, K., Haddadi, H., Amar, Y., Mortier, R., Li, Q., Moore, J., et al., 2018. Building accountability into the internet of things: the iot databox model. Journal of Reliable Intelligent Environments 4, 39–55.

[30] D'Acquisto, G., Domingo-Ferrer, J., Kikiras, P., Torra, V., de Montjoye, Y.A., Bourka, A., 2015. Privacy by design in big data: an overview of privacy enhancing technologies in the era of big data analytics. arXiv preprint arXiv:1512.06000 .

[31] Dai, W., Doröz, Y., Polyakov, Y., Rohloff, K., Sajjadpour, H., Savaş, E., Sunar, B., 2017. Implementation and evaluation of a lattice-based key-policy abe scheme. IEEE Transactions on Information Forensics and Security 13, 1169–1184.

[32] DEFeND, . The defend project. https://www.defendproject.eu/. Online; accessed 15/01/2021.

[33] EDPS EUROPA, . Meeting the challenges of big data: A call for transparency, user control, data protection by design and accountability. https://edps.europa.eu/sites/edp/files/publication/15-11-19_big_data_en.pdf. Online; accessed 26/02/2020.

[34] EU, G., . General data protection regulation. https://eur-lex.europa.eu/eli/reg/2016/679/oj. Online; accessed 01/15/2020.

[35] Ferrara, P., Spoto, F., 2018. Static analysis for gdpr compliance., in: ITASEC.

[36] Ferreira, A., Muchagata, J., 2018. Tagubig-taming your big data, in: 2018 International Carnahan Conference on Security Technology (ICCST), IEEE. pp. 1–5.

[37] Fischer-Hübner, S., Angulo, J., Karegar, F., Pulls, T., 2016. Transparency, privacy and trust–technology for tracking and controlling my data disclosures: Does this work?, in: IFIP International Conference on Trust Management, Springer. pp. 3–14.

[38] GDPR for Cloudera, . Compliance without complexity. https://www.cloudera.com/solutions/lower-business-risks/general-data-protection-regulation.html. Online; accessed 15/01/2021.

[39] Gjermundrød, H., Dionysiou, I., Costa, K., 2016. privacytracker: a privacy-by-design gdpr-compliant framework with verifiable data traceability controls, in: International Conference on Web Engineering, Springer. pp. 3–15.

[40] Goldman, K., Perez, R., Sailer, R., 2006. Linking remote attestation to secure tunnel endpoints, in: Proceedings of the first ACM workshop on Scalable trusted computing, pp. 21–24.

[41] Gonçalves, A., Correia, A., Cavique, L., 2019. An approach to gdpr based on object role modeling, in: World Conference on Information Systems and Technologies, Springer. pp. 595–602.

[42] Hashicorp, . Hashicorp vault. https://www.vaultproject.io/. Online; accessed 03/02/2020.

[43] Kneuper, R., 2020. Translating data protection into software requirements., in: ICISSP, pp. 257–264.

[44] Krempel, E., Beyerer, J., 2018. The eu general data protection regulation and its effects on designing assistive environments, in: Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference, pp. 327–330.

[45] Li, Z., Huan, S., 2018. Multi-level attribute-based encryption access control scheme for big data, in: MATEC Web of Conferences, EDP Sciences. p. 03047.

[46] Lopes, I.M., Guarda, T., Oliveira, P., 2019. Improvement of the applicability of the general data protection regulation in health clinics, in: World Conference on Information Systems and Technologies, Springer. pp. 155–165.

[47] Martin, Y.S., Kung, A., 2018. Methods and tools for gdpr compliance through privacy and data protection engineering, in: 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE. pp. 108–111.

[48] My data manager, . my data manager. https://www.mydatamanager.eu/conhecer-my-data-manager?lang=en. Online; accessed 03/02/2020.

[49] One trust, . One trust. https://www.onetrust.com. Online; accessed 03/02/2020.

[50] Oracle, . Addressing gdpr compliance using oracle data integration and data governance solutions. https://www.oracle.com/webfolder/s/delivery_production/docs/FY16h1/doc39/Addressing-GDPR-Compliance.pdf. Online; accessed 03/03/2020.

[51] Pandit, H.J., Debruyne, C., O'Sullivan, D., Lewis, D., 2018. An exploration of data interoperability for gdpr. International Journal of Standardization Research (IJSR) 16, 1–21.

[52] PAPAYA, . The papaya project. https://www.papaya-project.eu/. Online; accessed 15/01/2021.

[53] PDP4E, . The pdp4e project. https://www.pdp4e-project.eu/. Online; accessed 15/01/2021.

[54] Pedrosa, M., Costa, C., Dorado, J., 2019. Gdpr impacts and opportunities for computer-aided diagnosis guidelines and legal perspectives, in: 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS), IEEE. pp. 616–621.

[55] Peras, D., 2018. Guidelines for gdpr compliant consent and data management model in ict businesses, in: 29th International Conference of Central European Conference on Information and Intelligent Systems.

[56] Perfect, P., . Privacyperfect. https://www.privacyperfect.com/fr. Online; accessed 01/15/2020.

[57] Pham, P.l., 2019. The applicability of the gdpr to the internet of things. Journal of Data Protection & Privacy 2, 254–263.

[58] Piras, L., Al-Obeidallah, M.G., Pavlidis, M., Mouratidis, H., Tsohou, A., Magkos, E., Praitano, A., Iodice, A., Crespo, B.G.N., 2020. Defend dsm: a data scope management service for model-based privacy by design gdpr compliance, in: International Conference on Trust and Privacy in Digital Business, Springer. pp. 186–201.

[59] PoSeID-on, . The poseid-on project. https://www.poseidon-h2020.eu/. Online; accessed 15/01/2021.

[60] Rantos, K., Drosatos, G., Demertzis, K., Ilioudis, C., Papanikolaou, A., Kritsas, A., 2018. Advocate: a consent management platform for personal data processing in the iot using blockchain technology, in: International Conference on Security for Information Technology and Communications, Springer. pp. 300–313.

[61] Rhahla, M., Abdellatif, T., Attia, R., Berrayana, W., 2019a. A gdpr controller for iot systems: application to e-health, in: 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), IEEE. pp. 170–173.

[62] Rhahla, M., Allegue, S., Abdellatif, T., 2019b. A framework for gdpr compliance in big data systems. 14 the International Conference on Risks and Security of Internet and System (CRiSIS),Hammamet, Tunisia (2019) .

[63] Ringmann, S.D., Langweg, H., Waldvogel, M., 2018a. Requirements for legally compliant software based on the gdpr, in: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Springer. pp. 258–276.

[64] Ringmann, S.D., Langweg, H., Waldvogel, M., 2018b. Requirements for legally compliant software based on the gdpr, in: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Springer. pp. 258–276.

[65] Roy, S., Chuah, M., 2009. Secure data retrieval based on ciphertext policy attribute-based encryption (CP-ABE) system for the DTNs. Technical Report. Citeseer.

[66] Sabt, M., Achemlal, M., Bouabdallah, A., 2015. Trusted execution environment: what it is, and what it is not, in: 2015 IEEE Trustcom/BigDataSE/ISPA, IEEE. pp. 57–64.

[67] Sawant, N., Shah, H., 2013. Big data application architecture, in: Big Data Application Architecture Q & A. Springer, pp. 9–28.

[68] Schneider, G., 2018. Is privacy by construction possible?, in: International Symposium on Leveraging Applications of Formal Methods,

Springer. pp. 471–485.

[69] Schulz, K., Karovič, V., Veselỳ, P., 2021. Options to improve the general model of security management in private bank with gdpr compliance, in: Developments in Information & Knowledge Management for Business Applications. Springer, pp. 343–370.

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

[71] Skyhigh networks, . Skyhigh networks. https://www.skyhighnetworks.com. Online; accessed 03/02/2020.

[72] SMOOTH, . The smooth platform. https://smoothplatform.eu/. Online; accessed 15/01/2021.

[73] Tang, M., Shao, S., Yang, W., Liang, Y., Yu, Y., Saha, B., Hyun, D., 2019a. Sac: A system for big data lineage tracking, in: 2019 IEEE 35th International Conference on Data Engineering (ICDE), IEEE. pp. 1964–1967.

[74] Tang, M., Shao, S., Yang, W., Liang, Y., Yu, Y., Saha, B., Hyun, D., 2019b. Sac: A system for big data lineage tracking, in: 2019 IEEE 35th International Conference on Data Engineering (ICDE), IEEE. pp. 1964–1967.

[75] Tesfay, W.B., Hofmann, P., Nakamura, T., Kiyomoto, S., Serna, J., 2018. Privacyguide: Towards an implementation of the eu gdpr on internet privacy policy evaluation, in: Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics, pp. 15–21.

[76] Tsohou, A., Magkos, M., Mouratidis, H., Chrysoloras, G., Piras, L., Pavlidis, M., Debussche, J., Rotoloni, M., Crespo, B.G.N., 2019. Privacy, security, legal and technology acceptance requirements for a gdpr compliance platform, in: Computer Security. Springer, pp. 204–223.

[77] UK, I., . Data protection by design and default. https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/accountability-and-governance/data-protection-by-design-and-default/?q=necessary. Online; accessed 01/15/2020.

[78] Vault, A., . Alien vault usm. www.alienvault.com. Online; accessed 01/15/2020.

[79] Yuan, B., Li, J., 2019. The policy effect of the general data protection regulation (gdpr) on the digital public health sector in the european union: An empirical investigation. International journal of environmental research and public health 16, 1070.