Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 176 (2020) 2989–2998

Procedia Computer Science

www.elsevier.com/locate/procedia

24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Towards evaluating GDPR compliance in IoT applications

Christos Karageorgiou Kaneen, Euripides G.M. Petrakis*

School of Electrical and Computer Engineering, Technical University of Crete (TUC), Chania, Crete, Greece

Abstract

The General Data Protection Regulation (GDPR) was created for regulating how organizations that collect personal data process and protect it. In cases of digital handling of personal data, GDPR compliance must be proven by analyzing the actions that a system applies in order to gather, process and safeguard the data. We advocate that compliance must be considered in the design phase of the system, by analyzing the dependencies between system entities (e.g. personal data, users etc.) and the processes enacted upon them. Then, it is possible to generate a series of data reports that can be assessed by regulators who inspect the system for GDPR compliance. However, there can not be a universal methodology that covers all application domains and systems. To show proof of concept, we apply the methodology to a remote patient monitoring service that runs in the cloud.

Keywords: GDPR; Data protection; IoT; Property graph;

1. Introduction

Proving GDPR compliance [1] requires extensively documenting the procedures involving personal data, by evaluating their lawfulness and laying out the security measures enforced for its protection. It involves tackling complex issues such as (a) how personal data is stored, secured, processed, transmitted and erased, (b) how the policies related to its retention are enforced and (c) documenting proof of consent for its attainment and usage. This has led to the question of how can an organization assess the compliance of a system that handles personal data by considering the interactions between its components and the personal data it processes on behalf of its users.

GDPR requirements must be taken into account in the design phase of a system. Even so, the decision can not be reached automatically (e.g. an algorithm). Instead, compliance (or non-compliance) decisions are taken by experts known as Data Protection Officers (DPOs). This could be a subjective process, depending primarily on how different system component and process interactions are documented and on the ways personal data are handled. We advocate that essential GDPR compliance requirements can be incorporated into the system's class diagram. In this work, GDPR compliance requirements are defined and expressed as 10 fundamental questions whose answers are provided by querying the instantiated system. Some requirements involve different entities (e.g. users, databases) and address

* Corresponding author. Tel.: +030-28210-37229 ; fax: +030-28210-37542.
E-mail address: petrakis@intelligence.tuc.gr

1877-0509
10.1016/j.procs.2020.09.204

different functionality (e.g. storage, security), while others implicate more complex interactions between system and GDPR-related entities. The construction of the class diagram, is formulated as an incremental process with each step building on previously obtained results. In order to facilitate machine readability, the class diagram is transformed to a property graph. Then, compliance can be verified by applying a series of 10 queries on the system's property graph (related to the aforementioned requirements). This allows for generating data-filled reports that can be used by DPOs for evaluating compliance with the standard.

The method favors simplicity over completeness by focusing on an minimum set of essential GDPR obligations and provides a flexible but straightforward means for designing a system which embodies regulation entities necessary for aiding compliance evaluation. The resulting class diagram is a query-able structure for generating data reports related to each compliance requirement. As there can not be a universal methodology that covers all application domains and systems, the same process must be followed for different systems. To show proof of concept, we apply the methodology to a reference architecture of a remote patient monitoring service [2] that is general-purpose, highly modular, expandable and runs in the cloud. It is secure by design by also considering important privacy-awareness principles. The original design is remodeled to embed GDPR requirements. Following the proposed compliance approach, objects and their associations are mapped to graph nodes and relationships between them. The data-filled reports result from answering the 10 compliance queries applied on a graph model instantiated with simulated realistic data. Possible cases of GDPR compliance and non-compliance are discussed.

Sec. 2 presents work related to GDPR and system design. A reference architecture which is used for demonstrating the proposed methodology is discussed in Sec. 3. Sec. 4 shows how GDPR requirements are incorporated into system design and Sec. 5 shows how a system model instantiated with data can be evaluated for compliance with the standard. Conclusions and issues for future research are discussed in Sec. 6.

2. Related work

Recently, Neo4j[1] has attempted to connect personal data across all system components (i.e. services and databases) and track where and how personal information is stored, how it is used, how it moves to different systems and locations, who has access to this data and whether data owners have provided (or revoked) consent for doing so. However, no solution to the GDPR data protection problem is proposed or, if any, it is proprietary and undisclosed.

Torre et al. [3] propose a generic conceptual model for the GDPR using UML and describe an approach to check for compliance using the Object Constraint Language. Tom et al. [4] propose a preliminary model of GDPR concepts for helping system designers better understand associations within the regulation. However, they do not show how these rules may be extracted. Robol et al. [5] present a modeling language (that extends STS-ml) for representing the relationships between personal data processing entities. Ayala-Rivera et al. [6] describe a model-based approach to help organizations understand the data protection obligations imposed by the GDPR (Art. 5 and 25). Despite illustrating a high-level approach for complying to data protection obligations, the authors do not provide a methodology for evaluating compliance with other aspects of the regulation. Tamburri [7] details the results of applying formal concept analysis (FCA) to the GDPR by describing conformance clauses and rules to adhere to when processing data with the goal of extracting key insights for aiding the re-design of systems. Although, similar to one of our work's aims in helping system designers achieve compliance, we consider GDPR requirements in the initial design phase, adhering to a more flexible system design methodology.

Kammueller et al. [8] integrate privacy access control mechanisms in the design of an IoT healthcare patient monitoring system. The authors break down the regulation into requirements and show how to combine the so-called "Decentralized Labeling Model (DLM)" with Fusion and UML methods in order to produce a GDPR compliant IoT architecture. Unlike our work, personal data requirements relating to where the personal data is stored, whether consent was provided, data usage enabled by consent type and tracking of the personal data, are not considered at all. Consent and Data Management Model[2] is an on-going project at Trinity College, Dublin, that aims at utilizing Semantic Web technologies and providing a framework for supporting GDPR compliance in system design. GDPR-tEXT is an RDF representation of the GDPR's text (at article-paragraph granularity) and a vocabulary of relevant terms and concepts, GConsent is an ontology for representing consent for GDPR compliance, etc. In an application of this technology, Pandit, O'Sullivan and Lewis, [9] suggest that adherence to GDPR compliance can be answered

[1] https://neo4j.com/use-cases/gdpr-compliance/
[2] https://openscience.adaptcentre.ie/projects/CDMM/

by appropriately documenting and semantically annotating system processes. The project attempts to take into consideration the entire regulation requirements and provide an automated means for checking compliance by applying a test-driven approach. Due to the evolving, yet ambitious and fragmented nature of the work, more concrete results are yet to be seen (they address a limited, consent-specific set of requirements).

3. Use Case

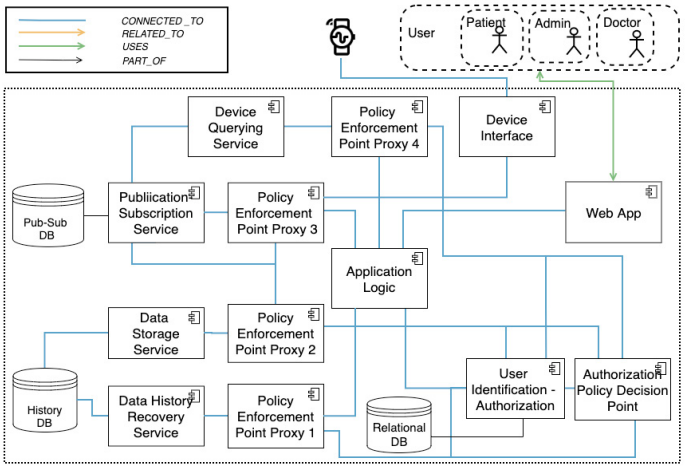

Fig. 1: Remote patient monitoring system.

We apply the methodology to a remote patient monitoring system that is a re-design of the users and functional requirements of iXen [2], a general purpose IoT reference architecture and system that runs in the cloud. The reason for selecting iXen as a use case lies in the flexibility of its design and security features: iXen is interoperable and expandable (i.e. services can be added or removed including services for enforcing GDPR) while being secure by design: all services are protected by an OAuth2.0[3] mechanism so that it is easier to justify that data is protected (i.e. access to data is granted only to authorized users or services based on roles and access policies). To better adapt this architecture to privacy by design [10] and data protection by design [11] principles, we consider data encryption at-rest and in-transit mechanisms as inherent to all services, databases, applications and devices. Moreover, all system users are covered by an access control mechanism, as further discussed. Fig. 1 illustrates iXen's components architecture (i.e. supported services and their interconnection).

The following user types are supported: (a) *patients*, who wear heart monitoring sensor devices that send biometric data to the *sensors connector* service, (b) *patient carers* (e.g doctors), who subscribe to a number of patients in order to monitor their conditions and are entitled to access patient data and (c) *administrators* who hold the highest degree of access-right privileges (i.e. they are granted access to user data and system services).

The system supports the following types of services:

1) Sensor services: Heart-rate-monitoring IoT devices are connected and continuously transmit data to the *Device Interface* service. Sensors register to the *Publish-Subscribe*[4] service that users and other services can subscribe to for getting notified on value changes or, when new values become available.

2)Storage services: The *Publish-Subscribe* service employs a MongoDB database that holds information about the active devices transmitting data, current device data (e.g. recently transmitted data) and active subscriptions. Heart-related health data, as received from devices and past (i.e. history) data is stored in the *History database* service (another MongoDB) using the *Data Storage* service that includes Cygnus[5]. The *Data History Recovery* service parses and translates user queries to equivalent MongoDB queries. Information related to user login credentials and authorization, such as roles and access permissions, is stored in a relational database which is a component of the *User Identification and Authorization*[6] service.

[3] https://oauth.net/2/
[4] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Orion+Context+Broker
[5] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Cygnus
[6] https://keyrock.docs.apiary.io/#reference/keyrock-api/role

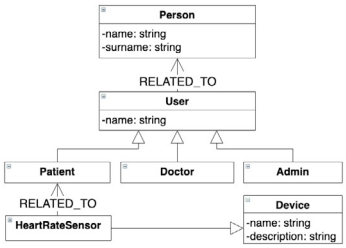

Fig. 2: Class diagram for Q0.

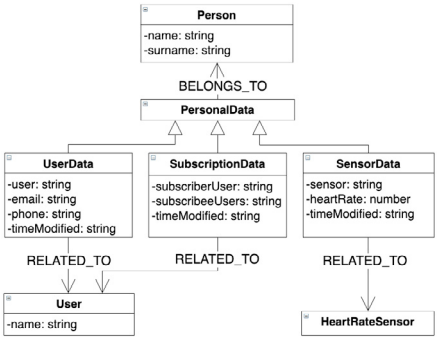

Fig. 3: Class diagram for Q1.

3) Application Logic: Orchestrates, controls and executes services in order. When a request is received (from a user or service), it is dispatched to the appropriate service. User requests are issued on the Web application interface. First, a user logs in to iXen using a user name and password. The user is then assigned a role (by the cloud administrator) and receives a token encoding their access rights (i.e. authorization to access iXen services). This is a responsibility of the *User Identification and Authorization* service.

4) Security services: Implement access control to services based on user roles and access policies. Once a user is logged-in, they are assigned an OAuth2 token encoding their identity. The respective user access rights are described by means of XACML[7]. For each user, a XACML file is stored in the *Authorization Policy Decision Point (PDP)*[8] service. Services offering a public interface are protected by a *Policy Enforcement Proxy*[9] service. Each public service is protected by a separate PEP service. It is the responsibility of this service to approve or reject a request to the protected service. The decision of whether the user is authorized to access the protected service will be determined by evaluating the XACML file. This process is carried out by the *Authorization PDP* service that responds to the PEP service with a decision. If the request is approved, it is forwarded to the protected service.

4. Incorporating GDPR Requirements into System Design

The following discussion guides the incremental construction of the system's class diagram using UML with reference to related GDPR articles. To understand the full life cycle of personal data in a system, we argue that this can be done by providing answers to a series of questions reflecting GDPR compliance requirements. In this work, we design a system class diagram based on the entities addressed by each question (e.g. the answer to each question adds the related entities to the class diagram). Details on this process can be found in [12]. The complete class diagram (for the system as a whole) can be viewed on the Web[10].

Q0: What are the fundamental data entities? This question lists data entities directly associated with personal data. The data entities can be (a) data owners and (b) device entities related to it. Fig. 2 illustrates the respective class diagram. Class *Person* denotes a general entity representing the owners of personal data. These can be a *User*, with *Patient*, *Doctor* or *Administrator* being special types of users. The personal data are acquired by sensors connected to patients [GDPR Art. *4(1)*].

Q1: What is the personal data within the system? The categories of personal data are *UserData* (e.g. user name, email, phone), *SubscriptionData* (e.g. users publishing data and users subscribing to data) and *SensorData* (e.g. heart rate sensor data) [GDPR Art. *4(1,13,14,15)*]. Fig 3 is the respective class diagram.

Q2: Where is the data stored? According to its type, data is stored in three databases namely, *RelationalDB* for *UserData*, *PubSubDB* for *SubscriptionData* together with the time this data is obtained (e.g. from sensors) and *HistoryDB* holding the time series created from the history of data (e.g. past sensor measurements) [GDPR Art.*4(6)*]. This information introduces the entities illustrated in Fig. 4.

Q3: How and when was the data obtained? This relates to information about the entities providing data (e.g. sensors) and the time a explicit consent was provided by the data owner. This question introduces two new entities in

[7] https://fiware-tutorials.readthedocs.io/en/latest/administrating-xacml/index.html
[8] https://authzforce-ce-fiware.readthedocs.io/en/release-5.1.2/
[9] https://fiware-pep-proxy.readthedocs.io/en/latest/
[10] http://www.intelligence.tuc.gr/~petrakis/downloads/final.png

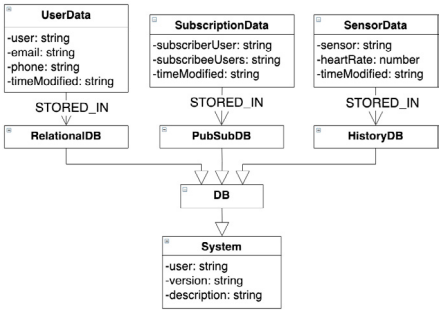

Fig. 4: Class diagram for Q2.

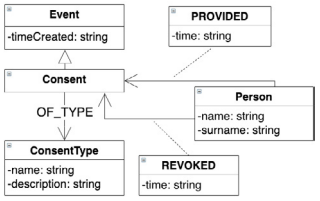

Fig. 5: Class diagram for Q3.

the class diagram of Fig. 4 relating to a *Consent* event provided (or revoked) by a *Person* entity and, the *ConsentType* describing the type of the consent (e.g. "Terms and Conditions") [GDPR Art. *4(11)*].

Q4: What is the need for this data? The class diagram of Fig. 6 represents the usage of data that a consent enables and, the categories of events that process personal data. The corresponding entities are *DataUsage* (the purpose of data processing which is enabled by a specific *ConsentType*) and *DataProcess*, representing an event of data processing, involving personal data belonging to a person. The latter introduces the classes (a) *DataAccess*: the event of data access to a *User* from a Web Application (class *WebApp*), (b) *DataUpdate*: an update operation issued by a Web Application to a database (class *DB*), (c) *DataExport*: the event of exporting data from a database to a third party service (*AmbulanceService* subclass of *ThirdParty*), (d) *InternalDataMovement*: the event of an internal data movement involving data exchanged between two systems (where class *System* can be a service or a database). *DB* and *Service* are sub-classes of *System*, that should not be confused with the remote patient monitoring system that contains it. *TO* and *FROM* associations indicate the flow between entities [GDPR Art. *4(2,4,10,11), 6, 7, 30*].

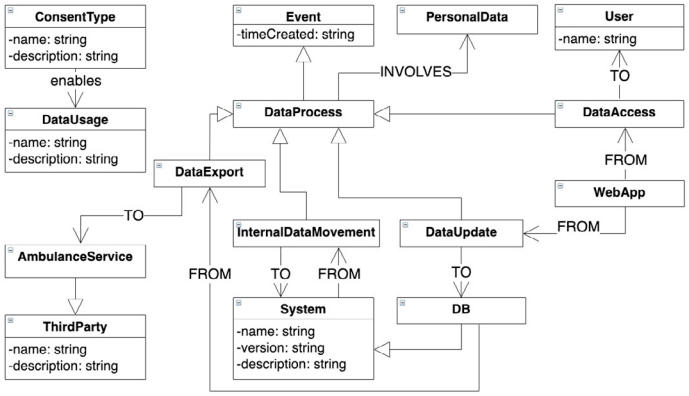

Fig. 6: Class diagram for Q4.

Q5: Who has access to the data? The class diagram of Fig. 7 represents new information about processes and users for whom access to the data is granted. Data access rights are denoted by *CAN_ACCESS_DATA_OF* associations between users [GDPR Art. *15, 28*].

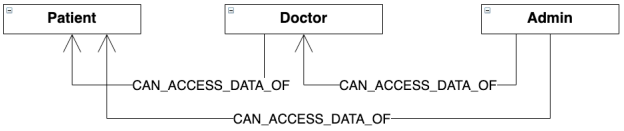

Fig. 7: Class diagram for Q5.

Q6: Did the data owner provide consent to use the data? The entities involved in the answer are *PersonalData*, *Person*, *Consent*, *ConsentType*, *DataUsage* which are already represented in the class diagram [GDPR Art.*7*]. of Fig. 4 and Fig. 5. No new entities are introduced in the class diagram.

Q7: Is the data secure? This question relates to security mechanisms enforcing data protection (e.g. OAuth2, encryption, etc.). The system supports OAuth2 security for all users and services and encryption for all systems and devices by design. In Fig. 8 the related entities are (a) *MasterKey*: a secret key for secure inter-service communication, (b) *Permission*: a permission for accessing a particular resource, (c) *Role*: a collection of permissions, described in its corresponding XACML rule, (d) *XACMLRule*: a rule formally describing the services and resources that users can access based on the roles and permissions they hold [GDPR Art. *4(5), 7, 32*].

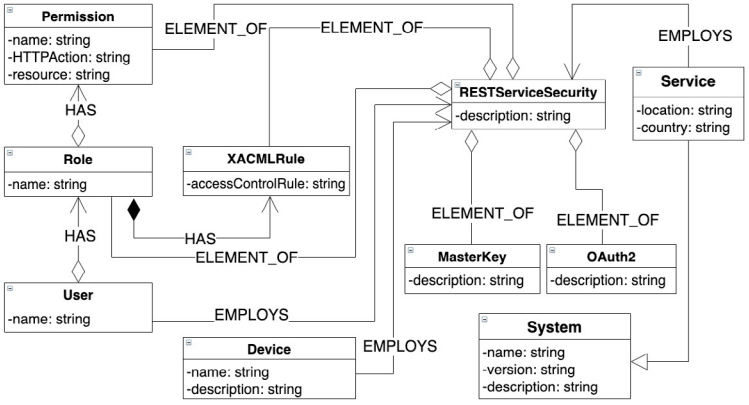

Fig. 8: Class diagram for Q7.

Q8: How does the data travel through systems? Provide information related to data lineage and data tracking (i.e. how and when personal data moves). This question relates to entities of Q4: *PersonalData*, *Person*, *Event*, *System*, *ThirdParty*, *App*, *User*, *ConsentType* and *DataUsage* [GDPR Art. *30*]. No new entities are introduced.

Q9: Does the data ever cross international borders? This question concerns the entities *PersonalData*, *Person*, *DataProcess*, *System*, *App*, *User*. [GDPR Art. *4(23), 44*]. No new entities are introduced.

5. GDPR Compliance Evaluation

The UML class diagram is now transformed to a graph using the mapping of Table 1. Classes are mapped to graph nodes and associations between classes are mapped to graph edges (relationships). Class attributes are translated to node properties and association names to relationship types. Class-subclass inheritance chains are represented by a node with the attributes of each sub-class as properties. As an example, Fig. 9 illustrates the mapping of the class diagram of Fig. 5 to property graph.

Table 1: Mapping class diagrams to property graphs.

UML	Property Graph
Class	*Node*
Class name	*Node label*
Association or Association class	*Relationship*
Association or Association class name	*Relationship type*
Attribute	*Property*
Class-subclass correlation	*Node with class and subclass names and attributes as labels and properties*

The answer to each of the 10 inquiries, referred to in Sec. 4, forms a separate report that corresponds to the system's response to a specific GDPR compliance requirement. These reports can be formally analyzed by a DPO for evaluating GDPR compliance. The actual queries are expressed in *Cypher*[11]. In [12] we show how the queries can also be expressed in a platform-independent graph query language which we developed for this purpose.

Querying the class diagram's property graph provides information regarding the abstract domain entities connected in the system. However, reaching GDPR compliance decisions requires answering questions regarding the details

[11] https://neo4j.com/developer/cypher-basics-i/

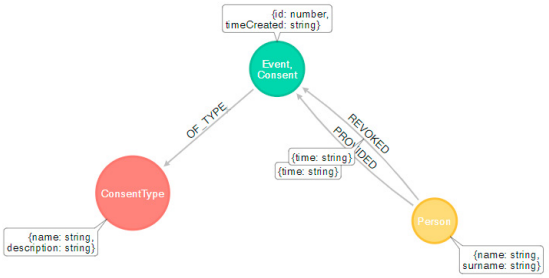

Fig. 9: Mapping the class diagram of Q3 to property graph.

of personal data (i.e. the instances of classes). Our application domain involves complex interactions e.g. between personal data, storage entities, third parties and cross-border processing events (questions 8 and 9) that may require explicit insight into data-specific information (exact location, country, time etc.). Therefore, compliance decisions can be made after careful examination not only of the domain model of the system but also through its data instances by observing the circulation of personal data in an intricate context. For instance, a DPO needs to know where a person's personal data is stored and how it is processed. If the types of information being collected and processed is not specified for all users or, information about how and when data was acquired or consent to use personal data has been provided for all users then, it is not possible to inspect the system based on class information alone. For example, a DPO might need to know, for each user, the time of the provided consent. Hence, since GDPR compliance demands meticulous data inspection and, most requirements are data-specific, in order for a DPO to take a decision for compliance (or non-compliance), the system under evaluation must be instantiated with real data. In the following, example data-filled reports which are returned in response to each query are presented, containing instantiated data. Because real-life data was not available to us, these were generated artificially.

Q0: What are the fundamental data entities? Table 2 is an example report illustrating device and user categories and their instances.

Table 2: Fundamental data entities.

type	fdDataEntity
[Device,HeartRateSensor]	{name:hrs, description:A sensor device that tracks your heart rate.}
[Person]	{name:Pat,surname:Williams}
[Person]	{name:Doc,surname:Brown}
[User,Admin]	{name:uAdam}

Q1: What is the personal data within the system? The answer reports personal data entities and the persons they belong to. Table 3 is an example illustrating a sensor measurement obtained from the heart rate monitor ("hrs") of patient Pat Williams and doctor Doc Brown subscribing to this information.

Table 3: Personal data in the system.

pdType	personalData	person
[PersonalData, SensorData]	{heartRate:100.0,timeModified:2018-12-11T20:08:13.539991+03:00,sensor:hrs}	{name:Pat, surname:Williams}
[PersonalData, SubscriptionData]	{subscriberUser:uDoc,timeModified:2018-12-11T16:01:16.645876+03:00, subscribeeUsers:[uPat]}	{name:Doc, surname:Brown}
[PersonalData, UserData]	{phone:+306900000003,timeModified:2018-11-25T10:12:03.448076+03:00,user:uAdam, email:adamsmith@domain.com}	{name:Adam, surname:Smith}

Q2: Where is the data stored? Reports the databases where personal data is stored. The report of Table 4 lists two databases, a MongoDB for sensor data for patient Pat Williams and a MySQL database for user profile data (this database is used for user authentication and authorization). Failing to document the database systems or adequate physical location information, heightens the risk for non-compliance (e.g. having null as a location or country result).

Q3: How and when was the data obtained? Reports information about consent provided by each user. Consent legitimises authorization and provides the legal basis demanded for data processing activities. It plays a crucial role

Table 4: System databases.

pdType	personalData	person	DB	location	country
[PersonalData, SensorData]	{heartRate:100.0, timeModified:2018-12-11T20:08:13.539991+03:00, sensor:hrs}	{name:Pat, surname: Williams}	{name:histMongoDB, country:GR, description:Heart Rate Sensor History DB,location:Heraklion, version:4.0.10}	Heraklion	GR
[PersonalData, UserData]	{phone:+306900000001,user:uPat,timeModified:2018-12-11T12:31:14.645876+03:00,email:patwilliams@domain.com}	{name:Pat, surname: Williams}	{name:MySQL, description:User Id & Authentication DB, version:8.0.16}	null	GR

towards fulfilling the legal obligations towards compliance. Table 5 illustrates information regarding the user or device fundamental data entity each personal data node is related to and consent that the data owner has provided (i.e. time of consent creation, time of consent provision and consent type). In the case were one of the consent information fields (e.g. consent type or time of provision) is undocumented, a GDPR regulator might deem the keeping of such data as unlawful, which would violate GDPR consent requirements[12].

Table 5: Information about user consent to use personal data.

pdType	personal Data	person	fdDataEntity	consent	consentType	timeConsent Provided
[PersonalData,SensorData]	(...)	(...)	{name:hrs, description:A sensor device that tracks your heart rate.}	{timeCreated:2018-12-11T12:32:13.645876+03:00}	{name: Terms & Conditions,description:General user usage terms and conditions.}	2018-12-11T12:32:13.645876+03:00
[PersonalData,UserData]	(...)	(...)	{name:uPat}	{timeCreated:2018-12-11T12:32:17.645876+03:00}	{name: Patient Emergency Condition,description:Patient health emergency situation condition.}	2018-12-11T12:32:17.645876+03:00

Q4: What is the need for this data? Reports information about the specific usages of each personal data entity (enabled by the consent provided by its owner), the data processing events that involve it (e.g. data access) by means of data processing and the entities (e.g. applications, systems, users, third parties) to and from which it travels (in the context of each data processing event). As per Art. 30, the GDPR obligates written documentation and recording of processing activities, which must be made completely available to authorities upon request. Inability to fully document and demonstrate such data processing activities leads to non-compliance[13]. Table 6 reports usages of personal data of user Pat Williams obtained from a heart rate sensor.

Q5: Who has access to the data? Reports access information for each personal data entity, including the data owner and all users authorized to access it. Answering this question becomes relevant for committing to GDPR's data protection and data privacy requirements (involving safety against unauthorized access). Being able to inspect which system users are authorized to access each personal data node reinforces the data owner's data rights. Failure to fully document such information may lead to non-compliance. Table 7 lists users with permission to access the sensor database (history DB) and the user database (user DB). Within the system, administrator Adam has been granted access to both databases while, doctor Doc Brown has access to the history database only.

Q6: Did the data owner provide consent to use the data? The answer reports information about the data usages enabled by the owners of each personal data entity and a boolean value denoting whether the data processors holding the data have permission to use it (which depends on whether consent for related data usages has been provided and not revoked). Processing permission builds upon *Q3*'s consent considerations also reflecting upon GDPR's "Right of Access" and "Right to be Forgotten" requirements. This answer may be a determinant for non-compliance if the *permissionToUseData* field of Table 8 evaluates to *false* for too many or sensitive personal data entities.

[12] https://gdpr.eu/gdpr-consent-requirements/
[13] https://gdpr-info.eu/issues/records-of-processing-activities/

Table 6: Personal data usages and processes in the system.

pdType	personalData	person	dataUsages	dpType	dataProcess	from	to
[PersonalData,SensorData]	(...)	(...)	[{name: Sensor Data Auto-sending, description:(...)}, {name:Personal Data Gen & Processing, description:(...)}, {name: Emergency User Data Export, description:(...)}]	[Event, DataProcess, Internal Data Movement]	{timeCreated:2019-04-08T09:01:03.508032+03:00}	[[System,Service,DeviceInterface], {name: Device Interface Service, country:GR, location: Heraklio}]	[[System,Service,PEPProxy3], {name: Wilma3, country: GR, description: (...),location: Heraklion}]
[PersonalData,UserData]	(...)	(...)	[{name: Sensor Data Auto-sending, description:(...)}, {name: Personal Data Gen & Processing, description:(...)}, {name:Emergency User Data Export, description:(...)}]	[Event, DataProcess,DataExport]	{timeCreated:2019-04-15T17:06:00.27773+03:00}	[[System,DB, RelationalDB],{name: MySQL, description: User Id & Authentication DB, version: 8.0.16}]	[[ThirdParty, AmbulanceService], {name: Emergency Ambulance Service, description:First aid emergency ambulance service.}]

Table 7: Users granted access to system databases.

pdType	personalData	person	DBType	DB	pdOwnerUser	otherUsersWAccess
[PersonalData,SensorData]	(...)	(...)	[System, DB,Service, HistoryDB]	(...)	{name:uPat}	[{name:uDoc}, {name:uAdam}]
[PersonalData,UserData]	(...)	(...)	[System, DB, RelationaDB]	(...)	{name:uDoc}	[{name:uAdam}]

Table 8: Consent provided by data owner.

pdType	personalData	person	dataUsagesEnabledByPerson	permissionToUseData
[PersonalData, SensorData]	{heartRate:100.0,timeModified:2018-12-11T20:08:13.539991+03:00,sensor:hrs}	{name:Pat, surname: Williams}	[{name:Emergency User Data Export,description: Export of personal patient data to an external, third party ambulance service (in case of an emergency health situation)}, (...)]	true

Q7: Is the data secure? Reports information about the type of security enforced within the system (REST Service Security), its elements (e.g. OAuth2.0) and a boolean value denoting whether each personal data entity is secure (which depends on whether the fundamental entity it is related to, employ REST service security). Since all users, devices and systems of the use case architecture employs REST service security, as well as encryption mechanisms (by default), the personal data is deemed secure. This answer may be a determinant for non-compliance if a human expert, such as a DPO, deems the security elements involved with each personal data entity as inadequate for fulfilling GDPR security demands (Art. 32 of GDPR).

Table 9: Information about data traveled.

pdType	personalData	person	eventType	eventDescription	time
[PersonalData, SubscriptionData]	(...)	(...)	[Event, DataProcess, DataUpdate]	FROM: MyHeartMonitor (App) | TO: MySQL (System)	2019-04-15T17:06:00.27773+03:00
[PersonalData, UserData]	(...)	(...)	[Event, DataProcess, DataExport]	FROM: MySQL (System) | TO: Emergency Ambulance Service (ThirdParty)	2019-04-15T17:06:00.27773+03:00
[PersonalData, UserData]	(...)	(...)	[Event, DataProcess, DataAccess]	FROM: MyHeartMonitor (App) | TO: uDoc (User)	2019-04-10T11:15:00.448076+03:00

Q8: How does the data travel through systems? The answer contains descriptions of each consent or data process event (e.g. consent types and data usages) related to data movements between systems and to the time each event occurred. Being able to provide key insight into how, when and where data flows in the system is a major factor towards proving that necessary measures are in place for meeting GDPR demands. Inadequate *eventDescription* field information, such as missing source or destination entities, or non-existent time information for some personal data entities may pose a risk for non-compliance. Table 9 results from querying the instantiated system graph about how data travels (i.e. information *FROM* and *TO* in *eventType* and *eventDescription* fields describes the process of data movement) and if consent was provided.

Q9: Does the data cross international borders? This answer, similar to *Q8*, may also be of interest to a DPO, for inspecting the countries from and to which personal data may travel (e.g. whether EU borders were crossed). As an example, a transfer done for processing purposes from a service of an EU controller to a non-EU third party service (e.g. cloud provider), might expose the former to severe compliance according to provisions of Chapter 5 of the regulation.

6. Conclusions and future work

This work addresses a methodology for system design by considering basic GDPR requirements of varying complexity in a step-by-step process and provides a means for querying the system for aiding compliance evaluation. Favoring ease in applicability, our work lacks in vigorous quantification of the whole breadth of the GDPR legal text. By focusing on a set of cornerstone requirements, we demonstrate a methodology towards achieving compliance evaluation for IoT applications of reasonable complexity. It is a best effort and constitutes work at an initial phase. There are also aspects of system design that still need to be addressed such as improving adherence to data protection principles (e.g. data minimization and third party storage). Investigating Semantic Web approaches for encoding system design information is also an interesting issue for research.

References

[1] E. Parliament, C. of European Union, Regulation (eu) 2016/679 of the european parliament and council (2016).
URL https://eur-lex.europa.eu/eli/reg/2016/679/oj

[2] X. Koundourakis, E. Petrakis, ixen: Content-driven service oriented architecture for the internet of things in the cloud, in: Intern. Conf. on Ambient Systems, Networks and Technologies (ANT 2020), Warsaw, Poland, 2020, pp. 145–152.
URL http://www.intelligence.tuc.gr/%7Epetrakis/publications/iXen.pdf

[3] D. Torre, G. Soltana, M. Sabetzadeh, L. C. Briand, Y. Auffinger, P. Goes, Using models to enable compliance checking against the gdpr: An experience report, in: ACM/IEEE 22nd Intern. Conf. on Model Driven Engineering Languages and Systems (MODELS 2019), Munich, Germany, 2019, pp. 1–11.
URL https://ieeexplore.ieee.org/document/8906896

[4] J. Tom, E. Sing, R. Matulevicius, Conceptual representation of the gdpr: Model and application directions, in: Intern. Conf. on Business Informatics Research (BIR 2018), Stockholm, Sweden, 2018, pp. 18–28.
URL https://link.springer.com/chapter/10.1007/978-3-319-99951-7_2

[5] M. Robol, M. Salnitri, P. Giorgini, Toward gdpr-compliant socio-technical systems: Modeling language and reasoning framework, in: IFIP Working Conference on The Practice of Enterprise Modeling (PoEM 2017), Leuven, Belgium, 2017, pp. 236–250.
URL https://link.springer.com/chapter/10.1007/978-3-319-70241-4_16

[6] V. Ayala-Rivera, L. Pasquale, The grace period has ended: An approach to operationalize gdpr requirements, in: IEEE 26th International Requirements Engineering Conference (RE 2018), Banff, AB, Canada, 2018, pp. 136–146.
URL https://ieeexplore.ieee.org/document/8491130

[7] D. A. Tamburri, Design principles for the general data protection regulation (gdpr): A formal concept analysis and its evaluation, Information Systems 91 (2019) 101469.
URL https://www.sciencedirect.com/science/article/pii/S0306437919305216

[8] F. Kammüller, O. O. Ogunyanwo, C. W. Probst, Designing data protection for gdpr compliance into iot healthcare systems (2019). arXiv:1901.02426.
URL https://arxiv.org/abs/1901.02426

[9] H. Pandit, D. O'Sullivan, D. Lewis, Test-driven approach towards gdpr compliance, in: Semantic Systems, Intern. Conf. on Semantic Systems (SEMANTiCS 2019), Springer International Publishing, Karlsruhe, Germany, 2019, pp. 19–33.
URL https://link.springer.com/chapter/10.1007/978-3-030-33220-4_2

[10] P. Guarda, N. Zannone, Towards the development of privacy-aware systems, Information and Software Technology 51 (2) (2009) 337–350.
URL http://www.sciencedirect.com/science/article/pii/S0950584908000578

[11] F. Blix, S. A. Elshekeil, S. Laoyookhong, Data protection by design in systems development: From legal requirements to technical solutions, in: 12th Intern. Conf. for Internet Technology and Secured Transactions (ICITST 2017), Cambridge, UK, 2017, pp. 98–103.
URL https://ieeexplore.ieee.org/document/8356355

[12] C. Karageorgiou-Kaneen, Methodology for designing gdpr compliant iot applications, Diploma thesis, Technical University of Crete (TUC), Chania, Crete, Greece (12 2019).
URL http://www.intelligence.tuc.gr/index.php?module=view&class=publication_file&id=591