Recommender-based Privacy Requirements Elicitation — EPICUREAN

An Approach to Simplify Privacy Settings in IoT Applications with Respect to the GDPR

Christoph Stach
University of Stuttgart, IPVS / AS
Stuttgart, Germany
stachch@ipvs.uni-stuttgart.de

Frank Steimle
University of Stuttgart, IPVS / AS
Stuttgart, Germany
steimlfk@ipvs.uni-stuttgart.de

ABSTRACT

Due to the *Internet of Things* (*IoT*), a giant leap towards a *quantified self* is made, i. e., more and more aspects of our lives are being captured, processed, and analyzed. This has many positive implications, e. g., *Smart Health* services help to relieve patients as well as physicians and reduce treatment costs. However, the price for such services is the disclosure of a lot of private data. For this reason, Smart Health services were particularly considered by the *European General Data Protection Regulation* (*GDPR*): a data subject's *explicit consent* is required when such a service processes his or her data. However, the elicitation of privacy requirements is a shortcoming in most IoT privacy systems. Either the user is overwhelmed by too many options or s/he is not sufficiently involved in the decision process. For this reason, we introduce *EPICUREAN*, a r**e**commender-based **priv**a**c**y req**ui**rements **e**licit**a**tio**n** approach. EPICUREAN uses modeling and data mining techniques to determine and recommend appropriate privacy settings to the user. The user is thus considerably supported but remains in full control over his or her private data.

CCS CONCEPTS

• **Security and privacy** → **Privacy protections**; **Usability in security and privacy**; *Distributed systems security*; *Database and storage security*; *Software and application security*; • **Information systems** → *Association rules*; *Clustering*; Collaborative filtering.

KEYWORDS

Privacy requirements elicitation, recommender system, knowledge modeling, clustering, association rules, privacy system, IoT, eHealth

ACM Reference Format:
Christoph Stach and Frank Steimle. 2019. Recommender-based Privacy Requirements Elicitation — EPICUREAN: An Approach to Simplify Privacy Settings in IoT Applications with Respect to the GDPR. In *The 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19), April 8–12, 2019, Limassol, Cyprus.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3297280.3297432

1 INTRODUCTION

Mark Weiser described the *computer for the 21^st^ century* as an omnipresent companion which we do not even take notice of [38]. In this way, it provides users with constant support in any situation. This vision has become reality due to the *Internet of Things* (*IoT*). In the IoT, everyday objects—e. g., watches, cellphones, or even cars—are interconnected. These devices are equipped with various sensors which capture a wide range of data about their users' lives. By combining these individual aspects and sending them to a high-performance processing back-end, a lot of information can be derived and valuable knowledge is gained. Therefore, there are many different application scenarios for the IoT including *Smart Homes*, *Smart Traffic*, and *Smart Health* [23].

In particular, Smart Health services are on the rise. The reason for this trend is that physicians are relieved and treatment costs can be significantly reduced, as patients can carry out a lot of the required medical examinations using telemedical techniques. For this purpose, smart health devices help patients to perform measurements and forward the gathered data to a medical data processing back-end. There, the data is analyzed, and the results are presented to the physicians [4]. Smart Health is highly flexible and can be used in many different situations, e. g., as a baby monitor, virtual health coach, emergency assistant, or continuous health monitor [29]. Smart Health is especially suitable for monitoring and treating patients with chronic diseases, as they must carry out routine check-ups and treatment procedures on a regular basis [21]. Not only physicians and patients benefit from such services but also care providers, researchers, and insurance companies [36].

However, this implies that many different stakeholders need to access the health data. Therefore, data privacy and security are very important issues for Smart Health. Yet, a review of current electronic health record systems shows that these systems do not provide appropriate data protection measures [12]. One of the key issues are human errors, i. e., an unintended misconfiguration of the privacy system by non-technical users [14]. Not only since the *European General Data Protection Regulation* (*GDPR*) [9] came into force, a systematic privacy approach is required that can be applied equally to all Smart Health services. For this, the focus must be on the patients, i. e., the data subjects. The privacy approach must not only enable the patients to specify their privacy requirements in a simple manner and verify the system's compliance [15], but it has to improve the patient's awareness of privacy vulnerabilities and assist them in finding appropriate privacy settings as well [37].

SAC '19, April 8–12, 2019, Limassol, Cyprus

ACM ISBN 978-1-4503-5933-7/19/04…$15.00
https://doi.org/10.1145/3297280.3297432

PATRON[1] is a privacy system for IoT applications which maximizes the functionalities of an application while minimizing the disclosed private data [31]. Yet, it lacks a simple user interface as a great deal of manual intervention by domain experts is required in the configuration process (see Section 2.3 for more information). Therefore, we introduce a r**e**commender-based **pri**va**c**y req**ui**rements **e**licit**a**tio**n** approach[2], called *EPICUREAN*[3]. EPICUREAN automatizes the configuration process of PATRON as far as possible and provides users with privacy setting suggestions tailored to their needs. To this end, we make the following five contributions:

(1) We establish a three-phase privacy requirements elicitation process called EPICUREAN. **(2)** We introduce a hierarchical modeling technique to describe what knowledge can be derived from which raw data (*preparation phase*). **(3)** We introduce a process to specify privacy settings and learn which privacy settings are relevant for which user (*training phase*). **(4)** We introduce a process to find privacy settings fitting to the users' requirements and suggest further privacy settings (*application phase*). **(5)** We integrate EPICUREAN into PATRON (although we integrate it into PATRON, the concepts can be transferred to any arbitrary privacy system).

The remainder of this paper is as follows: Section 2 introduces a real-world Smart Health use case, discusses in which respect such applications are affected by the GDPR, and describes how PATRON helps to protect the user's privacy. Then, we derive from literature requirements towards a privacy system with respect to privacy requirements elicitation in Section 3. Section 4 introduces our recommender-based privacy requirements elicitation approach and details on its phases. Section 5 assesses EPICUREAN and compared to related work before Section 6 concludes this paper.

2 APPLICATION SCENARIO

Chronic Obstructive Pulmonary Disease (*COPD*) is a treatable obstructive lung disease characterized by an airflow limitation. If not treated properly, the symptoms caused by COPD worsen over time. According to the World Health Organization, COPD is one of the leading deaths causes worldwide. Therefore, Smart Health approaches to support the treatment of these patients are subject of many research projects. In Section 2.1, a Cloud-based Smart Health approach is introduced, which deals with data collection, data analysis, and data provisioning. What needs to be ensured in such an approach regarding the GDPR is the subject of Section 2.2. Lastly, PATRON is introduced in Section 2.3 and it is shown to what extent PATRON already fulfills the requirements resulting from the GDPR and which open issues remain in this respect.

2.1 ECHO — A Smart Health System for COPD

The *ECHO* project provides a Smart Health system to support COPD patients. To this end, a patient Smartphone application for data acquisition, a physician web application for data visualization, and a back-end for data processing (the *ECHO Platform*) are provided [4, 34]. The overall ECHO architecture is shown in Figure 1 in an abstracted representation. Data from various sources such as the ECHO patient application (e. g., manually entered health data or location data captured with the Smartphone's GPS receiver) as well as data from medical devices (e. g., the *Peak Expiratory Flow*) [33], are sent to the ECHO Platform via its *Health API*.

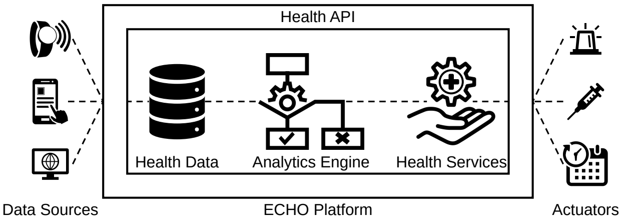

Figure 1: Overall ECHO Architecture.

Any incoming data is stored in the internal database, e. g., to repeat a certain analysis with altered parameters. The data is then processed by the *ECHO Analytics Engine*. This engine scans the data for specific patterns. A pattern is a sequence of certain sensor data and / or user input. For instance, one of these patterns can be "sputum color = green & PEF ≤ 7". If such a pattern is detected, then the integrated *Health Services* are notified. The Health Services can trigger events on actuators (e. g., to control the supply of medication), send messages (e. g., to warn patients about health changes), or visualize analysis findings for physicians.

Especially for the treatment of COPD it is important to know about the patients' activities to interpret medical results correctly. Therefore, there is a lot of research on how wearable sensors can be used to detect activity patterns and thus enable a long-term monitoring [5]. It is evident, why privacy has an important role in this context. In addition to all the medical data, ECHO has permanent access to the patients' location as well as activities and it can derive further knowledge from this data. In the following we discuss how such a service can comply with the GDPR.

2.2 Health Data in the Context of the GDPR

The introduction of GDPR had significant consequences for all domains. Any kind of data which provides insights into a data subject's "physical, physiological, genetic, mental, economic, cultural or social identity" [9] is covered by the GDPR. This also apply to any kind of health service such as ECHO. Additionally, the GDPR even addresses health data as well as the provisioning of health services, which reveal insights about a patient's health status.

Article 6 of GDPR defines six criteria which must be met by such a service, to consider its processing of health data to be lawful: a) The data subject must give consent to the data processing. b) It is necessary for the performance of a contract to which the data subject is party. c) It is necessary for compliance with a legal obligation. d) It is necessary to protect the vital interest of the data subject. e) It is necessary for the performance of a task carried out in the public interest. f) It is necessary for the purposes of the legitimate interests pursued by a third party.

For services processing predominantly health data, even higher standards are applied. Therefore, in Article 9 the GDPR explicitly prohibits the processing of health data unless the following criteria apply which sharpens Article 6: g) The data subject must give

[1] PATRON is an acronym for **P**riv**a**cy in S**tr**eam Pr**o**cessi**n**g.
[2] We introduce the concept of EPICUREAN. Prototypes are available for its components, but a total integration into PATRON as well as user studies are part of future work.
[3] Epicureanism is the doctrine for achieving the highest satisfaction while preventing any harm—i. e., it describes the key objective of PATRON.

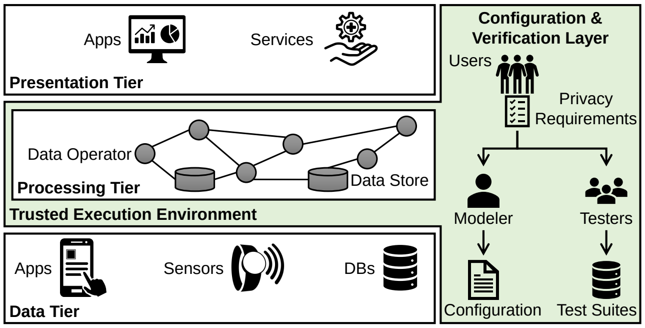

Figure 2: The PATRON Architecture [based on 31, 32].

explicit consent to the data processing. h) It is necessary for the purposes of preventive or occupational medicine. i) It is necessary for reasons of public interest in the public health domain.

This indicates that the GDPR aims at heavily involving patients in the data approval process. Although there is no distinct definition given, what *explicit consent* means, Article 4 defines *consent* as any action by a data subject that implies agreement to the requested processing of his or her private data. As compliance with these criteria must be ensured for every health service, a privacy-aware execution environment for such services is required. PATRON is such an execution environment. Next, we outline PATRON and assess whether it satisfies the GDPR.

2.3 PATRON — Privacy in Stream Processing

PATRON addresses the conflicting interests of users concerning privacy and service quality. That is, if a user shares no data with a service, his or her privacy is guaranteed. However, the experienced service quality is almost non-existent. Whereas if s/he shares all data with a service, it provides the best service quality, but privacy is at risk. Current privacy approaches focus on attribute-based data access control, i. e., a user must decide which service should get access to which data source. However, this is a very restrictive strategy. For instance, a user must decide whether ECHO is allowed to analyze his or her activities. While this is not a problem for medically relevant activities, it must not be allowed for leisure activities. For this reason, PATRON introduces patterns, i. e., sequences of high-level events, e. g., the patient did sports (Event$_1$) and then fell into a fit of coughing (Event$_2$). *Private patterns* are concealed while *public patterns* are shared with services. As a result, services can be provided with a maximum of data without compromising privacy.

As shown in Figure 2, PATRON consists of two components, a horizontal *Trusted Execution Environment* and a vertical *Configuration & Verification Layer* (green parts). In the following these two components are described with the focus on the configuration process. For more information, please refer to literature [24, 31, 32].

Trusted Execution Environment. The Trusted Execution Environment separates the processing of data (*Processing Tier*) completely from data acquisition (*Data Tier*) and data presentation (*Presentation Tier*). This enables PATRON to monitor all incoming data and regulate all outgoing data. If a private pattern is found in the incoming data, it is concealed from the execution logic running on the *data operators*. Multiple concealing techniques are available, e. g., events can be suppressed, obfuscated or reordered. Depending on the current situation, a quality metric selects the technique that produces the least *false positives* and *false negatives* in terms of public patterns (i. e., the one enabling the best service quality) [24]. In addition to real-time data, historical data can be protected likewise.

Configuration & Verification Layer. For the elicitation and specification of privacy requirements, PATRON applies a concept based on system theory. A user formulates *privacy requirements* in natural language and gives them to domain experts. These experts analyze the user's requested services for potential privacy risks and transform the privacy requirements accordingly to public and private patterns. This step is performed by several experts independently from each other. One of them is selected as the *modeler* and his or her patterns are used as the *configuration* of the Trusted Execution Environment. The remaining experts are considered as *testers* and their patterns are kept as *test suites* for subsequent validation. For the validation, the Trusted Execution Environment forwards the output of the Processing Tier to the Configuration & Verification Layer. It is analyzed whether private patterns have been disclosed (i. e., additional private patterns are required) or the service quality is insufficient (i. e., the private patterns are too restrictive). This is done by comparing the actual output with the one from the test suites [31]. Although the pattern generation is computer-assisted, a lot of work must be done manually, e. g., the experts need to contribute which knowledge can be derived from which data sources.

3 REQUIREMENTS SPECIFICATION

This application scenario shows that although the GDPR regulates Smart Health services, obtaining the explicit consent to data processing from the user becomes particularly difficult. While PATRON provides good technical mechanisms to regulate data access, its configuration—i. e., the involvement of the users—needs to be simplified. Therefore, we extracted the following requirements towards an elicitation mechanism for privacy requirements from literature:

R$_1$ The specification of privacy requirements must be very easy and practicable even for laymen [6].

R$_2$ As there are different perceptions of what is privacy-relevant, the users' requirements must be addressed individually [11].

R$_3$ Users must be made aware of further potential privacy threats they did not considering [8].

R$_4$ Privacy requirements must be categorizable to remain manageable [16].

R$_5$ The elicitation process must be comprehensible [10].

R$_6$ No third parties should be involved in the privacy requirements elicitation process, as they pursue other interests [17].

Taking these six requirements into account, an automated privacy requirements elicitation process is introduced in the following, which can be used in a privacy system such as PATRON.

4 EPICUREAN — RECOMMENDER-BASED PRIVACY REQUIREMENTS ELICITATION

It is obvious that PATRON currently does not fulfill these requirements. Although the specification of the privacy requirements is

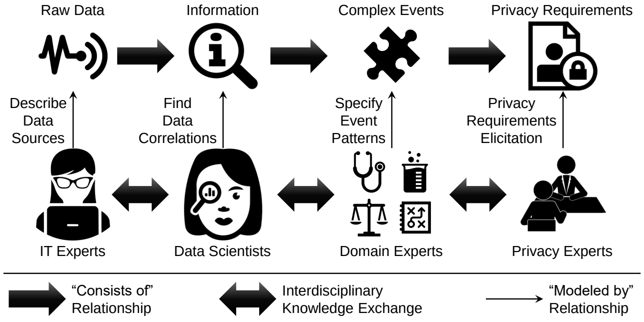

Figure 3: The EPICUREAN Knowledge Modeling Process.

very simple ($\mathbf{R_1}$) as well as individualized for each user ($\mathbf{R_2}$), the process of how patterns are derived from them is incomprehensible ($\mathbf{R_5}$) and cannot be made without the help of experts ($\mathbf{R_6}$). Moreover, users get no feedback, so they cannot anticipate all privacy threats originating from a certain service ($\mathbf{R_3}$), and they cannot manage their patterns at all ($\mathbf{R_4}$). To this end, we introduce EPICUREAN, a recommender-based privacy requirements elicitation process, that meets all these requirements.

The basic concept of EPICUREAN is inspired by *supervised learning*. In an initial preparation phase (see Section 4.1) experts from various domains analyze, annotate, and describe all data sources which are used by a certain service. This includes also which (potentially) privacy-relevant knowledge can be derived from these sources. EPICUREAN provides a comprehensive modeling technique for this purpose. Then, in the training phase (see Section 4.2) users specify their privacy requirements as normal, i. e., in PATRON in natural language. In this phase experts are still required to translate these requirements into patterns. EPICUREAN analyzes these requirements as well as the patterns and it establishes a knowledge base about the users. This knowledge base is used in the application phase (see Section 4.3) to derive fitting patterns from privacy requirements and even recommend additional patterns to the user. From then on, the elicitation of privacy requirements can be carried out fully automatically.

4.1 Preparation Phase

The aim of this initial phase is to persist the experts' knowledge. To this end, modelers from various domains describe private data at different levels of abstraction. This procedure is loosely based on the *DIKW-Pyramid*[4] [26]. Figure 3 shows how the four DIKW layers are adopted in EPICUREAN and identifies the key stakeholders:

Initially, *IT experts* identify and describe all data sources which are used by a certain service. The focus here is on *raw data*. That is, it is not considered which privacy threats originate from these sources, but rather which data is principally available to a service.

Data scientists are then able to analyze this data, reveal correlations within the data, and learn which high-order *information* can be derived from it. That is, this abstraction level describes how to interpret the raw data and initial privacy threats become noticeable.

Then, *domain experts* can use this information to specify patterns, i. e., *complex events*. These patterns already consider which actual values the information's attributes can have. The chronological sequence of individual events can be modeled as well. That is, in this abstraction level is specified which knowledge can be found in the data. It is not differentiated whether the knowledge is beneficial for the service quality or whether it represents a privacy threat—this must be decided for each user and service individually.

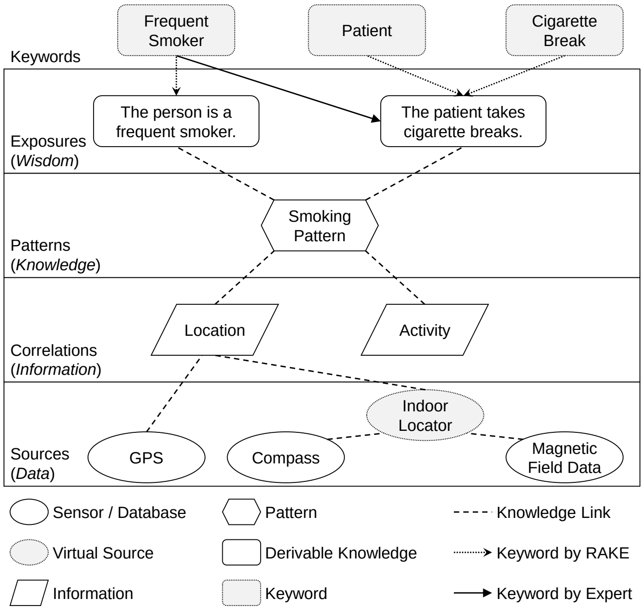

Figure 4: Excerpt of an EPICUREAN Knowledge Model for the ECHO Application Scenario.

Finally, *privacy experts* assign these patterns to *privacy requirements*. These privacy experts conduct many interviews with data subjects so that they can assess which privacy requirements are frequently formulated by them. Moreover, they are familiar with legal requirements that must be considered.

However, the modeling process does not end after a single pass. Instead, it is intended that the four expert groups exchange experiences and thus gradually refine and supplement their artifacts.

Figure 4 is an excerpt from a model for the ECHO use case (see Section 2.1). At the lowest level, all data sources used by the ECHO application are identified. In the given excerpt these are the *GPS receiver* and the *compass sensor*, which are all build into any current Smartphone, as well as a database with data about earth's *magnetic field*. Furthermore, so-called *virtual sources* can be modeled at this level as well. A virtual source is a combination of multiple (virtual) sources. In the example, an *indoor locator* is modeled as a virtual source, which determines the current (indoor) location of a Smartphone by combining compass data with magnetic field data (see [39]). At the next level, the information that can be derived from these (virtual) sources is modeled. For instance, the *location* of a data subject can be tracked using the GPS receiver or the indoor locator. In addition, ECHO also monitors the *activities* of a data subject[5]. These artifacts are used in the *smoking pattern*:

$$(location = smoking\ area) \longrightarrow (activity = moves\ hand\ to\ mouth)$$

[4] DIKW stands for **D**ata, **I**nformation, **K**nowledge, and **W**isdom.

[5] As Figure 4 is an excerpt, it is not shown from which sources the activity is derived.

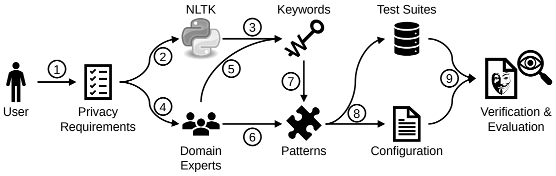

Figure 5: Workflow of the EPICUREAN Training Phase.

That is, specific values are assigned to the two artifacts and they are put into a temporal correlation—first the data subject enters the smoking area and then the cigarette is moved to the mouth. Finally, the highest level describes what wisdom is exposed when one of the patterns is discovered in the data[6]. The smoker pattern reveals that the data subject is a *smoker* and s/he takes *cigarette breaks*. Keywords can be assigned to each of these artifacts. They can be assigned either manually by the experts or automatically using natural language processing tools (see Section 4.2). These keywords are relevant for the application phase (see Section 4.3).

4.2 Training Phase

When the initial modeling is done, EPICUREAN can be brought on line. In this phase, however, the help of the experts is still required, as the system must gather knowledge about the users and learn about their privacy requirements. Nevertheless, as experts have access to the models from the previous phase and get tool support by EPICUREAN, their work is considerably simplified.

Figure 5 shows the workflow of the training phase. Initially, the user describes his privacy requirements in natural language as before (1). This description is analyzed by EPICUREAN (2). The *NLTK* (*Natural Language Toolkit*) [3] and the *RAKE* (*Rapid Automatic Keyword Extraction*) algorithm [25] are used for this analysis.

The procedure of the algorithm is shown in Listing 1. First, the input is split into individual tokens and converted into lowercase characters (Line 6). Then, the tokens are *lemmatized*, i. e., all words are reduced common base form (Line 8). For instance, '*am*', '*is*', and '*are*' are transformed to '*be*'. Please note that *stemming* is not suitable at this point, as it simply chops off the ends of words whereas lemmatization uses a vocabulary and does a morphological analysis of words. Although, no stop words must be removed, as the subsequent algorithm steps require the context of words and an intact structure of the sentences. The RAKE algorithm can then extract keywords from this cleansed data (Line 12). The keywords can also be ranked by the algorithm to find the most important ones (Line 13). The RAKE algorithm also works without the cleansing, but then the quality of the ranked results is worse[7].

```
1  import string
2  from nltk.stem import WordNetLemmatizer
3  from nltk.tokenize import word_tokenize
4  from rake_nltk import Rake
5  ### Preprocessing Input
6  tokens = word_tokenize(input.lower())
7  lemmatizer = WordNetLemmatizer()
8  lemmatized_tokens = [lemmatizer.lemmatize(t) for t in tokens]
9  clean_input = ' '.join(t for t in lemmatized_tokens)
10 ### Keyword Extraction
11 rake = Rake()
12 rake.extract_keywords_from_text(clean_input)
13 output = rake.get_ranked_phrases()
```

Listing 1: The EPICUREAN Keyword Extraction Algorithm.

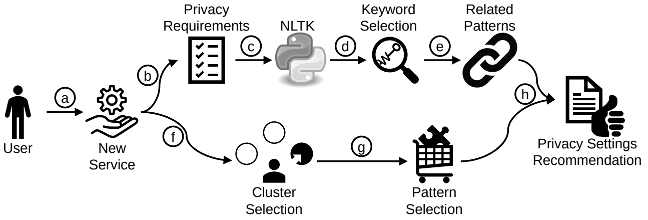

Figure 6: Workflow of the EPICUREAN Application Phase.

The extracted keywords are then linked to the privacy requirements (3). Experts also look at the privacy requirements (4) and tag them with keywords manually (5). Then, the domain experts create private patterns in accordance with the privacy requirements (6). Using the previously found keywords, the experts can search the model created in the preparation phase for further matching patterns which they did not have considered so far (7). Finally, a configuration and test cases can be created for PATRON in the same way as before. However, the mapping of the patterns to data sources is already available in the model (8). This enables an automatic creation of the configuration. The verification and evaluation of the patterns is also done as before in PATRON (9).

The privacy experts and domain experts have already modeled correlations between keywords, privacy requirements, and patterns in the preparation phase. However, these are based solely on experience and previous work. In the training phase, on the contrary, the models are based on privacy requirements of specific users for particular services. By no means, however, should the model of one phase replace that of the other phase, but the two phases complement each other to obtain a comprehensive knowledge base.

4.3 Application Phase

Once the model is reliable, the application phase can be started. In this phase, a kind of *collaborative filtering* is applied to recommend appropriate private patterns to users. This means that the behavior patterns of the users are analyzed and evaluated. Based on their behavior—i. e., their privacy requirements, their used services, their available data sources, and so forth—users are divided into clusters. Using the privacy requirements of these clusters, EPICUREAN infers the privacy requirements of each individual user. However, the recommendations are also tailored to the user's personal needs.

Figure 6 shows the workflow which is executed for this purpose. Whenever a user selects a new service (*a*), s/he can specify his or her privacy requirements in natural language (*b*)[8]. Like the training phase, EPICUREAN analyzes the privacy requirements and automatically assigns keywords to them (*c*). The knowledge base

[6] A privacy requirement would be the concealment of such a modeled exposure.

[7] More comprehensive keyword extraction mechanisms considering the application domain as well (e. g., *Pythia* [20]) can be applied if required.

[8] This applies also if a user changes the privacy requirements for an existing service.

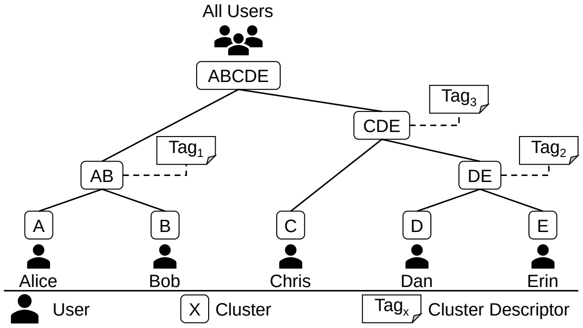

Figure 7: EPICUREAN’s Hierarchical Clustering.

is then scanned for matching keywords (d). For large knowledge bases (and therefore a huge list of modeled keywords), a simple equality check enough. For smaller data stocks, advanced similarity checks such as [2] can be used to find further patterns and thus make better recommendations to the user. Based on the found keywords, the model is browsed for appropriate patterns, i. e., patterns linked to similar keywords (e). Yet, this pattern list needs to be filtered to avoid overwhelming users with too many recommendations. The list also must be supplemented by patterns which the user has not considered when specifying the privacy requirements.

For this purpose, the collaborative filtering is applied. For the collaborative filtering, users must be divided into clusters, i. e., users in the same cluster are similar to each other in terms of privacy requirements whereas users in different clusters have heterogeneous privacy requirements (f). Unlike most cluster approaches, in which each element, i. e., each user, is assigned to exactly one cluster, EPICUREAN uses *hierarchical clustering*. To put it simply, a dedicated cluster is created for each user. Then, pairs of similar users are merged into new, shared clusters. This step is repeated, until all users are in a single large cluster. In this way, a hierarchy of clusters is created. To calculate the similarity of users, the *linkage criterion* is determined based on attributes such as quantity and type of services used or previous privacy requirements.

Figure 7 shows an example for hierarchical clustering as used in EPICUREAN. For each of the five users a dedicated cluster A to E is created. The user comparison indicates that Alice and Bob have the greatest similarity, as both refuse categorically to grant services access to their location. Therefore, a common cluster AB is created for them. This cluster is tagged descriptively, e. g., “Location-aware Users”. The clusters of Dan and Erin are merged into the tagged cluster DE accordingly, e. g., as they use Smart Health services, only. Since Chris has the greatest similarity to this cluster DE (e. g., all of them use a lot of services), a new tagged cluster CDE is created for these three users on the next hierarchy level. At the highest hierarchy level, all five users are merged into a collective cluster. Please note that neither the dedicated clusters nor the collective one is tagged—the composition of these clusters is self-explanatory.

Once it has been identified in which clusters the particular user belongs, the association analysis starts (g). The goal is to discover which privacy requirements similar users had and which patterns were applied. The resulting association rules look like this:

$$member(X, Cluster_\chi) \wedge uses(X, Service_\alpha) \Rightarrow has(X, PrivReq_\pi)$$

Table 1: Data Corpus for the Association Analysis.

Cluster	Service	User	Privacy Requirements
			$\dots$
CDE	$Service_\sigma$	C	$PrivReq_1, PrivReq_2, PrivReq_3$
CDE	$Service_\sigma$	D	$PrivReq_1, PrivReq_3$
CDE	$Service_\sigma$	E	$PrivReq_3$
			$\dots$

That is, a member of cluster χ who uses service α has the privacy request π. This type of rule requires expensive *multi-dimensional association rule mining*, as they involve more than one predicate (*member*, *uses*, and *has*). Due to the preceding clustering, however, these rules can be simplified, and *single-dimensional association rule mining* can be applied to each cluster in which the user is a member and each affected service. This results in this kind of rules:

$$has(X, PrivReq_\pi) \Rightarrow has(X, PrivReq_\varpi)$$

That is, EPICUREAN only must look cluster-wise for combinations of privacy requirements which are often applied to a service.

To illustrate this, an example is given in Table 1. Assuming Erin (E) uses $Service_\sigma$ for the first time and has the privacy requirement $PrivReq_3$. The given excerpt of the data corpus shows all privacy requirements of the users in cluster CDE for that $Service_\sigma$. To recommend further privacy requirements to Erin, EPICUREAN first determines all *frequent itemsets*. To this end, the *support* for each subset of occurring privacy requirements is determined. The support is defined as the fraction of users who apply the respective subset of privacy requirements. All support values are listed in Table 2. To suggest only relevant privacy requirements to the user, all item sets with a support of less than 50 % are removed. That is, EPICUREAN recommends Erin to use $PrivReq_1$ as well.

After the frequent itemsets for all clusters are determined, these results are combined with the results of the keyword search (h). The *privacy settings recommendations* are then created from these two sets. This way, not only private patterns corresponding to the entered privacy requirements are found, but also additional useful privacy requirements are shown to the user. All of this happens fully automatically, and experts are no longer required. EPICUREAN thus simplifies the privacy requirements elicitation substantially.

Table 2: Results of the Frequent Itemset Analysis.

Frequent Itemset	Support
$\{PrivReq_1\}$	66.6 %
~~$\{PrivReq_2\}$~~	33.3 %
$\{PrivReq_3\}$	100.0 %
~~$\{PrivReq_1, PrivReq_2\}$~~	33.3 %
$\{PrivReq_1, PrivReq_3\}$	66.6 %
~~$\{PrivReq_2, PrivReq_3\}$~~	33.3 %
~~$\{PrivReq_1, PrivReq_2, PrivReq_3\}$~~	33.3 %

5 ASSESSMENT

EPICUREAN is assessed in the following. First, we discuss whether it fulfills all identified requirements before we outline how our approach differs from a representative sample of related work.

Fulfillment of Requirements. In EPICUREAN, users describe their privacy requirements in natural language—the mapping to regulation of affected data sources is done automatically. Therefore, even laymen can address all their privacy concerns (**R$_1$**). Due to the hierarchical clustering, EPICUREAN is able to provide personalized recommendations for each user, i. e., a user receives privacy recommendations tailored to all clusters of which s/he is a member (**R$_2$**). These recommendations also make aware of further privacy requirements that might be relevant (**R$_3$**). Each privacy requirement is tagged with one or more keywords. These keywords categorize the privacy requirements (**R$_4$**). The clusters are also labeled. For each recommended privacy requirement, EPICUREAN is thereby able to show provenance information, e. g., for which cluster is the respective requirement intended and due to which property, the user was assigned to this cluster. Furthermore, collaborative filtering ensures that the amount of recommendations remains small and therefore manageable (**R$_5$**). Due to the knowledge base and the supervised learning approach, EPICUREAN operates fully automatically in the application phase (**R$_6$**). However, domain experts are still required for the preparation and training phase, as the understanding what knowledge can be derived from which data sources is highly domain-specific. Since the experts have no affiliation with the users or the services, it can be assumed that they make their decisions to the best of their knowledge and belief. Furthermore, their decisions are verified in PATRON by the reference group. Hence, EPICUREAN fulfills all identified requirements.

However, EPICUREAN gathers a lot of information about the user due to the usage of collaborative filtering. Regarding privacy, this seems to be not particularly persuasive. Yet, two aspects must be considered in this respect. On the one hand, EPICUREAN is part of the privacy mechanism itself. If a user cannot trust the privacy mechanism, then private data is at risk anyhow as it has unrestricted access to any sensitive data. On the other hand, there are privacy-aware approaches for hierarchical clustering [28] as well as approaches using *differential privacy* for collaborative filtering [40].

Related Work. LINDDUN [7] and *STPA-Priv* [27] enable developers to identify privacy threats in their services and apply privacy-enhancing technologies accordingly. To this end, threat scenarios are created for a specific application case. Yet, the solutions are only valid for the given cases and user-specific adaptations are not feasible. *PRET* [22] provides a general privacy requirements database and search tools, so developers can identify privacy risks in their services. All these approaches, however, assume benign developers who are willing to remove all privacy threats. By contrast, EPICUREAN provides technical support to the user to protect his or her private data without having to trust third parties.

ACCESSORS [30] is a modeling technique to describe correlations between raw data and derivable information on an abstract level. However, the models must be mapped to a specific application. Thus, the EPICUREAN knowledge model is superior to this approach due to its universal applicability and expressiveness.

Privacy Facets [35] is a framework to analyze privacy requirements learned from user studies to understand the users' privacy concerns. These concerns are reflected by privacy-critical information flow patterns. These patterns enable gap analyses for existing services and software requirements for future services can be derived. Yet, unlike EPICUREAN, these patterns are generic requirements that apply to all users and the software developers themselves are responsible for compliance with these privacy requirements.

Agarwal and Hall also target the identification of potential privacy leaks in services. Yet, their approach called *ProtectMyPrivacy* (*PMP*) [1] is based on crowdsourcing. Like PATRON, PMP monitors any running service and detects accesses to sensitive data. A user can decide at runtime whether s/he wants to grant a service the respective access permission or not. These decisions are sent to a PMP server. If a user is not sure whether s/he should grant a certain permission, the PMP analyzes the collected data and displays statistics on how most users have decided. However, contrary to EPICUREAN, the user has only attribute-based binary decision options (grant or deny access to a data source) and the provided recommendations are not tailored to him or her.

Lin et al. also focus on crowdsourcing. However, they use the swarm intelligence to verify the legitimacy of a data source access. Their *Privacy as Expectations* model [19] describes which access and authorization requests users consider to be justified for a service. This works for simple services and permission requests, e. g., when a navigation service requests access to a user's current location. The complex correlations between the variety of data sources used by many IoT services, as addressed by EPICUREAN, is not comprehensible and thus manageable for a common user without profound technical and domain knowledge.

Also, in social networks, users are overwhelmed by the configuration of their privacy settings. *SPAC* [18] uses machine learning techniques on existing privacy settings as well as the user's profile to recommend appropriate privacy settings for newly added friends. However, these recommendations are only as effective as the existing privacy settings—if the user was overwhelmed in the first place, this has a considerable impact on the quality of the recommendations. Ghazinour et al. use in their approach therefore also knowledge about typical privacy risks in social networks and carry out a cross-validation with other similar users [13]. However, these approaches only focus on social networks and their specific privacy threats, whereas EPICUREAN aims for any kind of service.

6 CONCLUSION

The IoT enables a variety of novel applications due to its comprehensive and continuous data capturing and processing. For instance, Smart Health services assist patients in dealing with their disease, as they can monitor their health condition and manage the medication on their own. To do this, however, the services need access to a lot of sensitive health data. Due to these potential privacy risks, the GDPR regulates in particular the handling of health data. Services need the data subject's explicit consent when processing health data. Approaches such as PATRON introduce a privacy-aware execution environment for such services. That is, it provides technical measures to guarantee the compliance of user-defined privacy requirements and therefore follows the *privacy by design* approach.

Yet, the elicitation of privacy requirements poses a challenge in such a system—either users are overwhelmed by too many options or they are not sufficiently involved in the configuration process. EPICUREAN addresses this issue: **(1)** It introduces a three-phase process supporting users in privacy requirements elicitation. **(2)** In the preparation phase, experts model what knowledge can be derived from which raw data. This model has four abstraction levels starting from a hardware-based description of available data sources to a high-level description of resulting privacy issues. **(3)** In the training phase, EPICUREAN uses this model to learn which privacy settings—i. e., which regulations must be applied to the data sources—are required to implement a given privacy requirement. **(4)** In the application phase, the learned model is used to implement a user's privacy requirements fully automatically. Moreover, the user gets personalized recommendations for additional privacy requirements that s/he did not consider. **(5)** EPICUREAN can be seamlessly integrated into PATRON's Configuration Layer. Nevertheless, its concepts can be transferred to any arbitrary privacy system. The assessment confirms that EPICUREAN not only fulfills all requirements for such a system with respect to the GDPR.

ACKNOWLEDGMENTS

This paper is part of the PATRON research project which is commissioned and funded by the Baden-Württemberg Stiftung gGmbH. The authors would like to thank the BW-Stiftung for this funding.

REFERENCES

[1] Yuvraj Agarwal and Malcolm Hall. 2013. ProtectMyPrivacy: Detecting and Mitigating Privacy Leaks on iOS Devices Using Crowdsourcing. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, 97–110.

[2] Prantik Bhattacharyya et al. 2011. Analysis of user keyword similarity in online social networks. *Social Network Analysis and Mining*, 1, 3, 143–158.

[3] Steven Bird et al. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc., Sebastopol, CA, USA. 502 pp.

[4] Marina Bitsaki et al. 2017. ChronicOnline: Implementing a mHealth solution for monitoring and early alerting in chronic obstructive pulmonary disease. *Health Informatics Journal*, 23, 3, 197–207.

[5] Bor-rong Chen et al. 2011. Long-term Monitoring of COPD Using Wearable Sensors. In *Proceedings of the 2nd Conference on Wireless Health*, 19:1–19:2.

[6] Erika Chin et al. 2012. Measuring User Confidence in Smartphone Security and Privacy. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, 1:1–1:16.

[7] Mina Deng et al. 2011. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16, 1, 3–32.

[8] Serge Egelman et al. 2013. Choice Architecture and Smartphone Privacy: There's a Price for That. In *The Economics of Information Security and Privacy*. Springer. Chap. 10, 211–236.

[9] European Parliament and Council of the European Union. 2016. Regulation on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (Data Protection Directive). Legislative acts L119. Official Journal of the European Union, (Apr. 27, 2016).

[10] Adrienne Porter Felt et al. 2012. Android Permissions: User Attention, Comprehension, and Behavior. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, 3:1–3:14.

[11] Adrienne Porter Felt et al. 2012. I've Got 99 Problems, but Vibration Ain't One: A Survey of Smartphone Users' Concerns. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, 33–44.

[12] José Luis Fernández-Alemán et al. 2013. Security and privacy in electronic health records: A systematic literature review. *Journal of Biomedical Informatics*, 46, 3, 541–562.

[13] Kambiz Ghazinour et al. 2013. Monitoring and Recommending Privacy Settings in Social Networks. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, 164–168.

[14] Arash Ghazvini and Zarina Shukur. 2013. Security Challenges and Success Factors of Electronic Healthcare System. *Procedia Technology*, 11, 2013, 212–219.

[15] Sebastian Haas et al. 2011. Aspects of privacy for electronic health records. *International Journal of Medical Informatics*, 80, 2, e26–e31.

[16] Zach Jorgensen et al. 2015. Dimensions of Risk in Mobile Applications: A User Study. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, 49–60.

[17] Ilias Leontiadis et al. 2012. Don't Kill My Ads!: Balancing Privacy in an Ad-supported Mobile Application Market. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, 2:1–2:6.

[18] Qingrui Li et al. 2011. Semantics-Enhanced Privacy Recommendation for Social Networking Sites. In *Proceedings of the 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, 226–233.

[19] Jialiu Lin et al. 2012. Expectation and Purpose: Understanding Users' Mental Models of Mobile App Privacy Through Crowdsourcing. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 501–510.

[20] Iouliana Litou and Vana Kalogeraki. 2017. Pythia: A System for Online Topic Discovery of Social Media Posts. In *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems*, 2497–2500.

[21] Farhaan Mirza et al. 2008. Mobile technologies and the holistic management of chronic diseases. *Health Informatics Journal*, 14, 4, 309–321.

[22] Seiya Miyazaki et al. 2008. Computer-Aided Privacy Requirements Elicitation Technique. In *Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference*, 369–372.

[23] Bill Montgomery. 2015. Future Shock: IoT benefits beyond traffic and lighting energy optimization. *IEEE Consumer Electronics Magazine*, 4, 4, 98–100.

[24] Saravana Murthy Palanisamy et al. 2018. Preserving Privacy and Quality of Service in Complex Event Processing Through Event Reordering. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, 40–51.

[25] Stuart Rose et al. 2010. Automatic Keyword Extraction from Individual Documents. In *Text Mining*. John Wiley & Sons, Ltd. Chap. 1, 1–20.

[26] Jennifer Rowley. 2007. The Wisdom Hierarchy: Representations of the DIKW Hierarchy. *Journal of Information Science*, 33, 2, 163–180.

[27] Stuart S. Shapiro. 2016. Privacy Risk Analysis Based on System Control Structures: Adapting System-Theoretic Process Analysis for Privacy Engineering. In *Proceedings of the 2016 IEEE Security and Privacy Workshops*, 17–24.

[28] Mina Sheikhalishahi and Fabio Martinelli. 2017. Privacy Preserving Hierarchical Clustering over Multi-party Data Distribution. In *Proceedings of the 10th International Conference on Security, Privacy, and Anonymity in Computation, Communication, and Storage*, 530–544.

[29] Dan Siewiorek. 2012. Generation Smartphone. *IEEE Spectrum*, 49, 9, 54–58.

[30] Christoph Stach and Bernhard Mitschang. 2018. ACCESSORS: A Data-Centric Permission Model for the Internet of Things. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 30–40.

[31] Christoph Stach et al. 2018. How a Pattern-based Privacy System Contributes to Improve Context Recognition. In *Proceedings of the 2018 IEEE International Conference on Pervasive Computing and Communications Workshops*, 238–243.

[32] Christoph Stach et al. 2018. The AVARE PATRON: A Holistic Privacy Approach for the Internet of Things. In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications*, 372–379.

[33] Christoph Stach et al. 2018. The Privacy Management Platform: An Enabler for Device Interoperability and Information Security in mHealth Applications. In *Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies*, 27–38.

[34] Frank Steimle et al. 2017. Extended provisioning, security and analysis techniques for the ECHO health data management system. *Computing*, 99, 2, 183–201.

[35] Keerthi Thomas et al. 2014. Distilling Privacy Requirements for Mobile Applications. In *Proceedings of the 36th International Conference on Software Engineering*, 871–882.

[36] Alastair van Heerden et al. 2012. Point of care in your pocket: a research agenda for the field of m-health. *Bull World Health Organ*, 90, 5, 393–394.

[37] Bryan Watson and Jun Zheng. 2017. On the User Awareness of Mobile Security Recommendations. In *Proceedings of the SouthEast Conference*, 120–127.

[38] Mark Weiser. 1991. The Computer for the 21st Century. *Scientific American*, 265, 3, 94–104.

[39] Hongwei Xie et al. 2016. A Reliability-Augmented Particle Filter for Magnetic Fingerprinting Based Indoor Localization on Smartphone. *IEEE Transactions on Mobile Computing*, 15, 8, 1877–1892.

[40] Xue Zhu and Yuqing Sun. 2016. Differential Privacy for Collaborative Filtering Recommender Algorithm. In *Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics*, 9–16.