The AVARE PATRON
A Holistic Privacy Approach for the Internet of Things

Christoph Stach[1], Sascha Alpers[2], Stefanie Betz[3], Frank Dürr[1], Andreas Fritsch[3], Kai Mindermann[4], Saravana Murthy Palanisamy[1], Gunther Schiefer[3], Manuela Wagner[5], Bernhard Mitschang[1], Andreas Oberweis[2,3] and Stefan Wagner[4]

[1]*IPVS, University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany*

[2]*FZI Forschungszentrum Informatik, Haid-und-Neu-Straße 10-14, 76131 Karlsruhe, Germany*

[3]*AIFB, Karlsruhe Institute of Technology, Kaiserstraße 89, 76133 Karlsruhe, Germany*

[4]*ISTE, University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany*

[5]*ZAR, Karlsruhe Institute of Technology, Vincenz-Prießnitz-Straße 3, 76131 Karlsruhe, Germany*

Keywords: Privacy, IoT Apps, Smart Things, Stream Processing, Privacy Preferences Elicitation & Verification.

Abstract: Applications for the Internet of Things are becoming increasingly popular. Due to the large amount of available context data, such applications can be used effectively in many domains. By interlinking these data and analyzing them, it is possible to gather a lot of knowledge about a user. Therefore, these applications pose a threat to privacy. In this paper, we illustrate this threat by looking at a real-world application scenario. Current state of the art focuses on privacy mechanisms either for Smart Things or for big data processing systems. However, our studies show that for a comprehensive privacy protection a holistic view on these applications is required. Therefore, we describe how to combine two promising privacy approaches from both categories, namely *AVARE* and *PATRON*. Evaluation results confirm the thereby achieved synergy effects.

1 INTRODUCTION

Applications for the *Internet of Things*—or short *IoT Apps*—are on the rise. This trend is due to the increasing number of sensors that are built in everyday objects with (indirect) connection to the Internet. So, these objects are able to interconnect and exchange data. Such devices are labeled as *Smart Things*. Since we are surrounded by Smart Things in almost every situation, there constantly arise novel application fields for the IoT. Here, each Smart Thing captures a different aspect of its context and commonly sends this data to a central processing system. The central system consolidates the data and has sufficient computing power to perform comprehensive analyses on it. The data is stored for future unknown purposes and gets enriched by data from further data sources. As a result, knowledge can be gained from the available data, e. g., behavior patterns can be derived for an individual user. This knowledge can be used to increase the quality of service for each user from then on.

GrowthEnabler predicts that the market for such IoT Apps will grow by almost 300 % until 2020. Especially the *Connected Health* domain benefits from this trend (GrowthEnabler, 2017). However, as health IoT Apps handle data which contains a lot of knowledge about users, an effective privacy system is a key issue for such apps. State of the art focuses on protecting sensitive data either on Smart Things or on big data processing systems. Yet, there is no holistic view on IoT Apps as a whole. That is, a privacy system for the IoT has to operate on both the Smart Things as well as the central processing system in order to provide a comprehensive protection. There are completely novel requirements towards such a privacy system that exceed those for single Smart Things or central processing systems by far (Dhillon and Backhouse, 2000). For instance, due to the complexity of IoT Apps, users cannot comprehend what knowledge can be gained from which data and what background information about him or her is available.

For this reason, by introducing *AVARE PATRON*, a holistic privacy approach which focuses on a simple description of privacy preferences, we make the following contributions: **(a)** We introduce a real-world application scenario from the Connected Health and derive requirements towards an IoT privacy system.

Stach, C., Alpers, S., Betz, S., Dürr, F., Fritsch, A., Mindermann, K., Palanisamy, S., Schiefer, G., Wagner, M., Mitschang, B., Oberweis, A. and Wagner, S.
The AVARE PATRON - A Holistic Privacy Approach for the Internet of Things.
DOI: 10.5220/0006850303720379
In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 2: SECRYPT*, pages 372-379
ISBN: 978-989-758-319-3

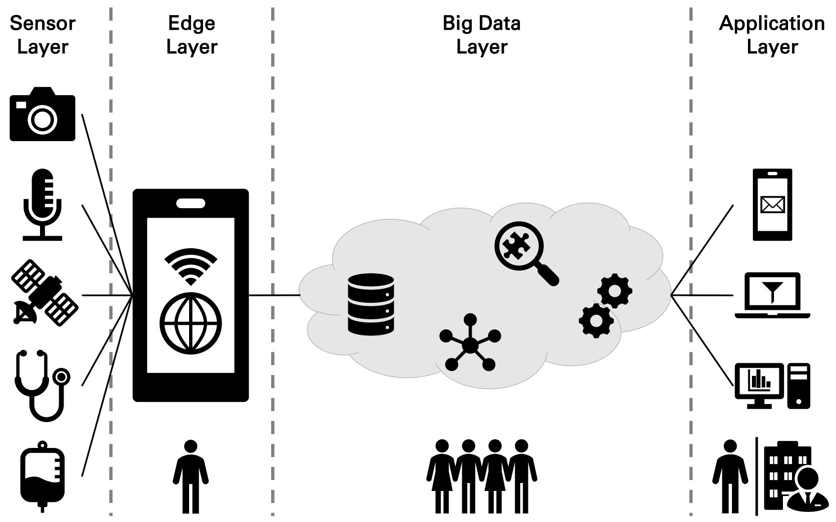

Figure 1: IoT App Layer Model.

(b) We introduce a holistic privacy approach for the IoT called AVARE PATRON, a fusion of two promising privacy systems for Smart Things (*AVARE*) and big data processing systems (*PATRON*). Both systems achieve significant synergy effects due to this tight intermeshing. **(c)** We show that AVARE PATRON is the only privacy approach which meets all requirements towards an IoT privacy system.

The remainder of this paper is as follows: In Section 2, a real-world IoT App from the Connected Health domain is introduced and privacy requirements are derived from it. Then, AVARE PATRON—our approach towards a holistic privacy system for the IoT—is introduced in Section 3. Section 4 discusses AVARE PATRON in the context of related work. Finally, Section 5 concludes the paper.

2 APPLICATION SCENARIO

Before introducing a real-world application scenario, we first discuss the architecture of IoT Apps in general. Figure 1 illustrates the components involved. The *Sensor Layer* contains all components that are able to capture context data. These components are characterized by the fact that they have very little computing power and can only capture a certain type of data. As a result, they only provide unprocessed raw data. They are also not designed to execute third-party applications. Examples of Sensor Layer components are cameras, microphones, or GPS receivers. Special medical devices such as health monitors or devices for the administration of medication are also part of this layer. In order to (pre)process the captured data, these devices have to be connected to a device with more computing power and connectivity (e. g., a smartphone). This connection can be either physical (e. g., the sensor is installed in a smartphone) or wireless (e. g., via Bluetooth). This hub device is usually strongly tied to a single user, i. e., all data conflated on the device can be linked to this user. Unlike sensors, devices in the *Edge Layer* are able to store data. Their key feature is however the ability to execute third-party applications. These applications have access to remote servers, i. e., they can transfer the collected data to the *Big Data Layer* in order to perform comprehensive analyses. As a result, data of many such hub devices, i. e., data of many users, are gathered in the Big Data Layer. A user does not know where his or her data is processed or stored. The Big Data Layer persistently stores any incoming data. This enables to process both, historical and real-time data. These two data types can be combined for the analyses. Data of different users can be cross-linked as well. That way, data mining, machine learning, and complex event processing techniques can be used to recognize patterns and generate further knowledge. The insights are prepared for presentations tailored to different stakeholders. For example, users can be informed about the occurrence of certain patterns, irrelevant information can be filtered out, or the data can be provided in a highly aggregated form. How the data is presented is determined by the *Application Layer*. The devices used here are highly heterogeneous and are not necessarily associated with the data subject, i. e., these devices can belong to other users or even companies.

After introducing the architecture of IoT Apps, the application scenario below illustrates the advantages of these apps. The scenario introduces an app for children suffering from diabetes, as IoT Apps are particularly effective for the treatment of chronic diseases (Knöll, 2012).

For this purpose, the children are equipped with a *Smart Bracelet* and a smartphone. The Smart Bracelet is able to detect certain activities (e. g., administering insulin) based on characteristic movement patterns (Kwapisz et al., 2011). The smartphone determines the child's mood via its microphone (Mehta et al., 2012) and the taken bread units via its camera (Almaghrabi et al., 2012). In addition, blood glucose meters can be connected via Bluetooth in order to access their readings. All measurement results are automatically captured, labeled with a GPS location and a time stamp, and entered into an electronic diabetes diary. Thereby the children do not have to remember to write down their health data and the records are more accurate.

The collected data is regularly sent to a hospital cloud. Here the data of all patients are preprocessed and made available to the attending physicians. This facilitates their work, as they can quickly review the data and focus on emergency cases. If necessary, however, they also have access to the unprocessed original data.

Additionally, this approach enables researchers from various research areas to gather novel insights due to the huge data collection. For instance, physicians can derive correlations between treatment methods and disease progressions. Or urban planners are able to identify places where the medical condition gets significantly better and learn how to create healthier cities (Knöll, 2012).

However, there are two further stakeholders for this kind of data: the parents of the young patients and their insurance companies. Parents can be informed about an unhealthy lifestyle of their children (e. g., if too many sweets are consumed) or be alerted in case of a sugar shock including precise location information. Insurance companies can use the data to determine whether expensive treatment methods are actually required.

Due to the huge amount of sensitive data and the large number of stakeholders who are interested in this data, a wide variety of privacy concerns arise in such a context. For instance, the children want to prevent that their parents are able to check their current location regularly and thus oversee them permanently. However, if they deactivate the GPS entirely, the parents cannot be informed of their whereabouts even in case of an emergency. Moreover, this would significantly deteriorate the data quality. Correlations between particular locations and health conditions can no longer be detected. This also applies to all other sensors used in this scenario. That is, an applicable privacy system cannot solely rely on filtering out or alternating data from certain sources.

Resulting Requirements.

Privacy systems for IoT Apps should be applied to the Edge Layer or the Big Data Layer. In the Sensor Layer, no third-party applications can be installed and these devices lack sufficient computing power. The data subject has no control over the devices in the Application Layer. In addition, a privacy system would have little effect in this layer; private data has already been fully analyzed and could have been shared with any third-party by previous layers. Therefore, the focus of this work is on the Edge Layer and the Big Data Layer.

R_1 Simple Configuration. Most users of IoT Apps are no IT experts. Therefore, the configuration of the privacy systems, i. e., the specification of privacy preferences, has to be very simple. This is crucial in such a complex environment, as users cannot comprehend what knowledge can be derived from which data. The user has to be able to give a high level description of which information should be concealed from which stakeholders.

R_2 Quality of Service Preservation. Besides privacy, the user is primarily interested in a high quality of service. As shown in the application scenario, all privacy issues could be solved by sharing no information with any third-party. But then the user would not benefit from the IoT App. Therefore, a privacy system has to apply different concealing techniques to protect privacy while still guaranteeing the highest possible service quality.

R_3 Protection at Big Data Layer. Only at the Big Data Layer all data used by an IoT App is available (i. e., data from all involved Smart Devices as well as data from further sources). In addition to real-time data, the Big Data Layer also has the necessary storage capacities to maintain a long-term data history. New knowledge can be derived by combining historical and real-time data. Therefore, a privacy mechanism needs access to all of this data to comply with the privacy preferences.

R_4 Protection at Edge Layer. As a user can no longer technically control the use of personal data after it has left the Edge Layer, it is necessary that a part of the privacy mechanism is also executed there. Since the amount of data involved in an IoT App is very large, data protection at the Big Data Layer takes a considerable amount of time. To ensure near-real-time data processing despite the privacy measures, the amount of data should be reduced at an early stage. For instance, a patient could prevent data from sensors that are not required for the IoT App from being transferred to the Big Data Layer. By restraining data at an early stage, it can also be ensured that particularly sensitive data never leaves the Smart Thing, i. e., the data subject's control.

R_5 **Tight Coupling.** The privacy mechanism at the Edge Layer and the one at the Big Data Layer should work closely together for efficiency reasons. The configurations of both mechanisms have to be harmonized according to the user's privacy preferences. This includes not only coordinating which mechanism protects what kind of data, but also a shared memory, which stores the existing knowledge on every component. It should be kept in mind here that a lot of different devices at the Edge Layer can be involved in a single IoT App.

3 THE AVARE PATRON

To meet these requirements, we combine two promising privacy approaches, namely *AVARE* (Alpers et al., 2017a) and *PATRON* (Stach et al., 2018). AVARE is a privacy mechanism for Smart Things. With AVARE, the user's effort in specifying his or her privacy preferences is reduced as s/he has to formulate these only once at a central point. AVARE then distributes them to all of his or her Edge Layer devices. Besides, legal compliance is taken into account. Nevertheless, the source-based permissions used in AVARE prevent comprehensive analytics in the Big Data Layer. PATRON takes care of the latter. It allows users to specify their privacy preferences in natural language. Domain experts are provided with tool support to semi-automatically translate these descriptions into permissions. For verification, a control group defines alternative permissions. Via a data flow comparison, the quality of the two permission sets is evaluated to verify that the permissions fully correspond to the user's preferences. By combining the two approaches, synergy effects are achieved in the resulting holistic privacy system. The overall architecture is shown in Figure 2. The components are outlined hereafter.

3.1 Configuration of the System

In addition to the existing four layers for IoT Apps, we introduce two further layers: the *Verification and Configuration Layer* and the *Deployment Layer*. These layers are required for creating and managing the privacy configurations.

The elicitation of privacy preferences has to be very easy, especially in such a complex environment as IoT Apps. The user therefore formulates his or her privacy preferences in near-natural language towards the PATRON *Verification and Configuration Layer*. Using a knowledge base, these preferences are translated into a system configuration. The knowledge base contains insights from domain experts (e. g., which data is required for which analysis or what knowledge can be derived from which sensor data). The translation can only be carried out semi-automatically, since the preferences are expressed at a high level (e. g., "Insurance company must not know that I have eaten sweets!"). System theoretical tools can support this process. Several configurations are created, of which one is randomly selected as master configuration. The remaining configurations are kept as test suites for verification (see Section 3.3).

The master configuration is sent to the *Deployment Layer*. If necessary, legal experts can assess the configurations for their legal compliance at this point. If the configuration does not violate any applicable law or norm, it is approved for deployment. For this, it is split into two: an AVARE configuration enforced directly on the various devices at the Edge Layer and a PATRON configuration for the Big Data Layer.

3.2 Privacy at the Edge Layer

The privacy system *AVARE* is embedded in the Edge Layer. This requires a dedicated instances of AVARE per Smart Thing on the Edge Layer equipped with a corresponding configuration, depending on the sensors available for the device.

So, the data can be prepared (i. e., thinned out) at the Edge Layer. AVARE is able to filter data horizontally or vertically (i. e., filtering values or attributes) blur data, and block data sources that are not required by later analyses. Depending on the respective data source, different concealing techniques are applied. For instance, it is acceptable to reduce the accuracy of location data (if this does not violate applicable law), while medical data must not be altered as this could have serious implications for a patient's health.

Such privacy mechanisms work entirely source-based, e. g., location data sharing can be prevented but patterns consisting of data sequences from different sources cannot be concealed. This could be achieved with a permission model such as ACCESSORS, but even then it is not possible to consider the data from other Smart Things. However, this data is merged in the Big Data Layer, whereby further patterns can be revealed. To provide a comprehensive protection, very restrictive rules are required at this layer. Yet, that has a negative effect on subsequent processing

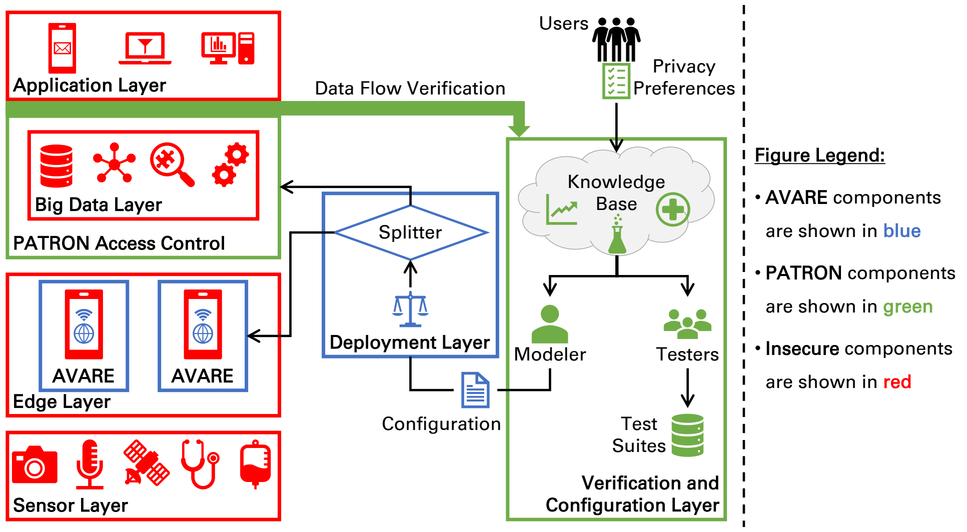

Figure 2: Overall Architecture of AVARE PATRON.

steps. The advantage of this procedure is that such privacy operations require little computing effort and by reducing the amount of data subsequent processing steps are much faster. Moreover, users do not have to rely on the fulfillment of their privacy preferences at the Big Data Layer. This means that the modelers have to trade off quality of service against processing speed in the Big Data Layer.

Additionally, the connection to the Big Data Layer can also be secured. Thereby devices in the Edge Layer cannot send data to unauthorized third parties, but to the processing back end, only. In addition, the connection itself can be protected (e. g., via SSL). This solves a major problem for PATRON, namely how to ensure that Smart Things do not bypass the *PATRON Access Control* and send their data directly to the Big Data Layer.

3.3 Privacy at the Big Data Layer

The PATRON Access Control is wrapped around the Big Data Layer. Therefore, it is able to control all incoming and outgoing data streams. That way, the PATRON Access Control has a detailed overview of the existing information available to the processing logic in the Big Data Layer.

PATRON does not filter out certain attributes, but it conceals private patterns. Such a pattern represents a sequence of high-level events, e. g., the patient ate sweets (Event_1) and had to inject insulin as a result (Event_2). To accomplish this, various techniques can be applied. For instance, the chronological sequence of events can be swapped (whereby the correlation between eating sweets and taking insulin gets lost) or certain events can be suppressed. On the one hand, the modeler has to consider which technique has the least influence on the quality of service. To this end, it is considered how many *false positives* (e. g., sequence $\text{Event}_2 \rightarrow \text{Event}_1$ is detected although it did not occur) and *false negatives* (e. g., an insulin intake is not detected) occur due to the manipulation. Depending on the specific use case, these parameters can also be weighted (e. g., a false alarm is less crucial than a non-identified emergency). On the other hand, the manipulation must not be perceptible. For instance, if an adversary knows that all n time units a certain reading occurs, this event can neither be suppressed nor swapped in the chronological order. As a result, the computations in the PATRON Access Control become complex and thus time-consuming. It is therefore important that the incoming data is pre-filtered in the Edge Layer by AVARE.

These concealing techniques are applied to both historical data and real-time data. The outgoing (privacy-friendly) data flow is not only forwarded to the Application Layer, but also sent to the Verification and Configuration Layer. Here, the data is compared with the result of a simulated run on which the test cases are applied. The system assesses whether the selected master configuration complies with all privacy preferences, as well as whether the configuration is too restrictive, i. e., whether the quality of service is impaired too much. The configuration is then adjusted accordingly and the user gets feedback. This is important to build confidence in the privacy system.

4 DISCUSSION

AVARE and PATRON are promising privacy systems in their respective application field. However, the combination of these two systems provides a holistic privacy protection for IoT Apps. In the following, we discuss whether the two systems meet the requirements towards a privacy system for IoT Apps specified in Section 2 and which synergy effects are achieved by our combined approach. As shown in the application scenario, the elicitation of privacy preferences has to be simple (**R$_1$**). Related work offer a variety of ways for the elicitation of privacy preferences. *ACCESSORS* introduces a permission model for Smart Things that allows users to describe which information can be used for which purpose under which context. It also enables to express what information can be derived from which sensor (combinations) (Stach and Mitschang, 2018). However, expert support is needed to identify these relationships. There are extensions for TOSCA (TOSCA is a standard for describing topology and orchestration of cloud applications) to specify privacy requirements (Breitenbücher et al., 2013). However, these preferences can be specified only by the system administrator and apply to all users. Yet, each user might have individual privacy preferences which cannot be considered in advance. In addition, system administrators and users have to trust blindly that these settings are actually implemented by the cloud provider. *STPA-Sec* introduces a method to model security requirements towards a software system. Thereby, software architects are semi-automatically supported to take these security constraints into account during implementation phase (Young and Leveson, 2014). *STPA-Priv* supports additionally modeling of privacy-relevant aspects (Shapiro, 2016). This presupposes that the architect is willing to protect private data—but a lot of business models are based on the exploitation of this data. In addition, STPA-Priv cannot apply any user-specific privacy preferences.

On the contrary, AVARE provides a user-friendly GUI that allows users to configure the system. It guides the user to create and manage a legally compliant privacy profile. This profile is then transferred and applied to all of his or her Smart Things. However, the user is not made aware of any correlations between the data sources used by an app and the knowledge derivable from these data. PATRON therefore adopts a different strategy. Since the user gives high-level descriptions of his or her preferences, s/he is able to formulate all of his or her requirements. As AVARE PATRON adopts this strategy, our approach fully meets **Requirement R$_1$**.

The data protection measures must not unnecessarily affect the service quality of the IoT App (**R$_2$**). To the best of our knowledge, this issue is not covered by related work. In AVARE, the user is responsible for maintaining the quality of service. That is, if the experienced quality of service is too low, the user has to decide which privacy settings has to be changed to improve the quality. PATRON always checks and maximizes the quality of service automatically. This is achieved by applying the most suitable concealing techniques for each individual case. However, the large number of possible techniques causes a processing overhead. In AVARE PATRON, this is reduced by aggregating or filtering out unrequired data at the Edge Layer. Thus, our approach is better than both of the other two approaches as it fully meets **Requirement R$_2$**.

Privacy protection at the Big Data Layer (**R$_3$**) is important in the IoT context. There are many privacy extensions for real-time data processing systems, e. g., *Borealis* (Lindner and Meier, 2006). They provide an attribute-based protection, i. e., certain data attributes are only visible to authorized processing units. This procedure is overly restrictive as the units either always have access to certain attributes or never. Therefore, systems such as *ACStream* provide context-based access control (Cao et al., 2009). Yet, they also operate at the level of attributes. He et al. therefore propose to consider whole complex events, i. e., sequential sequences of certain attribute values. This way, certain events can be dropped instead of all attributes. As a result, quality of service is considerably improved (He et al., 2011). Wang et al. study how to suppress certain events while maximizing service quality (Wang et al., 2013). However, dropping of events achieves suboptimal results, as too much information could be lost. To carry out accurate but privacy-aware analyses, *PrivApprox* uses a differential privacy-based approach. That is, detailed data from different users is analyzed, but the results contain no information about an individual user (Quoc et al., 2017). Yet, this technique is not suitable for medical use cases, since it is crucial that examination results can be linked to specific patients.

The Big Data Layer is fully out of AVARE's scope whereas PATRON fully meets this requirement. It provides comprehensive privacy protection at the Big Data Layer. Thus, AVARE PATRON also meets **Requirement R$_3$**.

Privacy protection at the Edge Layer (**R$_4$**) is vital in the IoT context as well. There are basically two different implementation strategies for privacy mechanisms for Smart Things: On the one hand, the apps can be manipulated. A monitoring component is injected into the byte code of an app. This monitor ensures that the app complies with the user's privacy preferences. *AppGuard* is a privacy system which uses byte code injection. It enables users to add context-based constraints to permissions, i. e., the permissions' scope of validity gets restricted. Furthermore, users are able to specify countermeasures if an app's execution violates the permissions. For instance, the app can be provided with aggregated, anonymized, or randomized data (Backes et al., 2014). *Dr. Android & Mr. Hide* addresses the problem that a lot of apps have access to too many private data due to coarse-grained permission settings. For this purpose, new fine-granular permissions are introduced. All of an app's functions that require user data are assigned to one of four protection classes. For each category, generic anonymization and filtering techniques are defined allowing the user to restrict data access (Jeon et al., 2012). Even more control is provided by *RetroSkeleton*. Here, the user can specify function calls s/he considers to be privacy critical. For each function call, s/he defines a code fragment which should be executed instead (Davis and Chen, 2013). Yet, users are completely overburdened with this task, e. g., as many approaches require a deeper technical understanding. On the other hand, the operating system on the Smart Things can be manipulated, i. e., the operating system monitors the apps. That way, similar extensions as the ones described above can be realized. For instance, *Apex* adds constraints to permissions (e. g., to specify how often an app can request the current location) (Nauman et al., 2010). However, all of these approaches have three crucial issues: a) As they map their permissions to sensors, privacy management is far too restrictive. b) They only work on single Smart Things and do not consider distributed infrastructures. c) Many of these approaches violate applicable law, e. g., the manipulation of byte code might violate copyright law (Alpers et al., 2017b).

Privacy protection at the Edge Layer is fully realized by AVARE. It provides various technical measures to protect the user's personal data on the Smart Things. This is ensured, e. g., by preventing unauthorized data access. PATRON does not consider privacy protection at the Edge Layer. By integrating AVARE in our approach, we also meet **Requirement $\mathbf{R_4}$** totally.

A holistic view on the entire data processing of an IoT App is highly recommended for a privacy system ($\mathbf{R_5}$). As related work considers only the one or the other, there is no such holistic view. AVARE PATRON operates on both layers, as there is a tight coupling between the privacy components at both layers. On the one hand, the two protection mechanisms are controlled by a common configuration. This enables a good coordination of concealing tasks. On the other hand, this ensures that Smart Devices on the Edge Layer forward their data to the PATRON Access Control, only. Thus, AVARE PATRON meets **Requirement $\mathbf{R_5}$**.

Lessons Learned.

Although AVARE and PATRON are highly effective in their respective application field, both lack a holistic privacy protection for IoT Apps.

AVARE has drawbacks with respect to quality of service. Since data protection is realized at data source level, the restrictions can be overly strict. This is a consequence of the fact that AVARE operates solely on Edge Layer devices. As a result, AVARE lacks the knowledge of how the data is processed in the Big Data Layer and with which additional data it is interlinked. As AVARE only has to deal with a small amount of data, it can provide near-real-time data processing.

That is the key problem of PATRON. The Big Data Layer is where the data from all Smart Things is gathered. As a result, PATRON has to deal with a lot of data. As PATRON operates at pattern level, there are many options for protecting private data. The selection and application of the best privacy mechanism is therefore very time-consuming. However, this ensures that the user receives the best possible quality of service.

Hence, only by combining the two approaches all requirements towards a privacy system for IoT Apps can be met. AVARE PATRON is not only the sum of both approaches' advantages, but it also eliminates their shortcomings. In addition, the tight coupling ensures that adversaries are unable to gain access to sensitive data. So, combining AVARE and PATRON is the logical choice.

5 CONCLUSION

Due to the proliferation of sensors in everyday objects, IoT Apps are becoming increasingly popular. Smart Things can be beneficial in various domains such as Smart Cities, Industrial IoT, and Connected Health. However, this also raises new requirements towards privacy mechanisms. While there is a large number of privacy solutions for Smart Things or for big data processing respectively, there is a lack of holistic approaches. For this reason, this paper describes how to combine two promising individual solutions, namely AVARE and PATRON. This combined approach is called AVARE PATRON. Evaluation results show that AVARE PATRON complies with all requirements towards a privacy system for IoT apps. AVARE PATRON not only possesses the individual advantages of its components, but also generates many synergies.

ACKNOWLEDGEMENTS

This paper is the result of a collaboration between the research projects AVARE and PATRON, both commissioned and financed by the Baden-Württemberg Stiftung gGmbH.

REFERENCES

Almaghrabi, R., Villalobos, G., Pouladzadeh, P., and Shirmohammadi, S. (2012). A novel method for measuring nutrition intake based on food image. In *Proceedings of the 2012 IEEE International Instrumentation and Measurement Technology Conference*, I2MTC '12, pages 366–370.

Alpers, S., Oberweis, A., Pieper, M., Betz, S., Fritsch, A., Schiefer, G., and Wagner, M. (2017a). PRIVACY-AVARE: An Approach to Manage and Distribute Privacy Settings. In *Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications*, ICCC '17, pages 1460–1468.

Alpers, S., Pieper, M., and Wagner, M. (2017b). Herausforderungen bei der Entwicklung von Anwendungen zum Selbstdatenschutz. In *Informatik 2017: Digitale Kulturen, Tagungsband der 47. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 25.9-29.9.2017, Chemnitz*, volume 275 of *LNI*, pages 1061–1072. *(in German)*.

Backes, M., Gerling, S., Hammer, C., Maffei, M., and Styp-Rekowsky, P. (2014). AppGuard — Fine-Grained Policy Enforcement for Untrusted Android Applications. In *Revised Selected Papers of the 8th International Workshop on Data Privacy Management and Autonomous Spontaneous Security - Volume 8247*, pages 213–231.

Breitenbücher, U., Binz, T., Kopp, O., Leymann, F., and Wieland, M. (2013). Policy-Aware Provisioning of Cloud Applications. In *Proceedings of the Seventh International Conference on Emerging Security Information, Systems and Technologies*, SECURWARE '13, pages 86–95.

Cao, J., Carminati, B., Ferrari, E., and Tan, K.-L. (2009). ACStream: Enforcing Access Control over Data Streams. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*, ICDE '09, pages 1495–1498.

Davis, B. and Chen, H. (2013). RetroSkeleton: Retrofitting Android Apps. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, pages 181–192.

Dhillon, G. and Backhouse, J. (2000). Technical Opinion: Information System Security Management in the New Millennium. *Communications of the ACM*, 43(7):125–128.

GrowthEnabler (2017). Market Pulse Report, Internet of Things (IoT). Report.

He, Y., Barman, S., Wang, D., and Naughton, J. F. (2011). On the Complexity of Privacy-preserving Complex Event Processing. In *Proceedings of the Thirtieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '11, pages 165–174.

Jeon, J., Micinski, K. K., Vaughan, J. A., Fogel, A., Reddy, N., Foster, J. S., and Millstein, T. (2012). Dr. Android and Mr. Hide: Fine-grained Permissions in Android Applications. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '12, pages 3–14.

Knöll, M. (2012). Urban Health Games. Collaborative, Expressive & Reflective. PhD thesis, University of Stuttgart.

Kwapisz, J. R., Weiss, G. M., and Moore, S. A. (2011). Activity Recognition Using Cell Phone Accelerometers. *ACM SIGKDD Explorations Newsletter*, 12(2):74–82.

Lindner, W. and Meier, J. (2006). Securing the Borealis Data Stream Engine. In *Proceedings of the 10th International Database Engineering and Applications Symposium*, IDEAS '06, pages 137–147.

Mehta, D. D., Zañartu, M., Feng, S. W., Cheyne II, H. A., and Hillman, R. E. (2012). Mobile Voice Health Monitoring Using a Wearable Accelerometer Sensor and a Smartphone Platform. *IEEE Transactions on Biomedical Engineering*, 59(11):3090–3096.

Nauman, M., Khan, S., and Zhang, X. (2010). Apex: Extending Android Permission Model and Enforcement with User-defined Runtime Constraints. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '10, pages 328–332.

Quoc, D. L., Beck, M., Bhatotia, P., Chen, R., Fetzer, C., and Strufe, T. (2017). PrivApprox: Privacy-Preserving Stream Analytics. In *Proceedings of the 2017 USENIX Annual Technical Conference*, ATC '17, pages 659–672.

Shapiro, S. S. (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*, SPW '16, pages 17–24.

Stach, C., Dürr, F., Mindermann, K., Palanisamy, S. M., and Wagner, S. (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*, CoMoRea '18, pages 238–243.

Stach, C. and Mitschang, B. (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*, ICISSP '18, pages 30–40.

Wang, D., He, Y., Rundensteiner, E., and Naughton, J. F. (2013). Utility-maximizing Event Stream Suppression. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 589–600.

Young, W. and Leveson, N. G. (2014). An Integrated Approach to Safety and Security Based on Systems Theory. *Communications of the ACM*, 57(2):31–35.