Towards an End-to-End Architecture for Run-time Data Protection in the Cloud

Published in the Proceedings of the 44th Euromicro Conference on Software Engineering and Advanced Applications

(SEAA), pp. 514-518, 2018

Nazila Gol Mohammadi, Zoltán Ádám Mann, Andreas Metzger, Maritta Heisel
paluno – The Ruhr Institute for Software Technology
University of Duisburg-Essen, Germany
Email: {firstname.lastname}@paluno.uni-due.de

James Greig
Oxford Computer Consultants
Oxford, United Kingdom
Email: james@oxfordcc.co.uk

***Abstract*—Protecting sensitive data is a key concern for the adoption of cloud solutions. Protecting data in the cloud is made particularly challenging by the dynamic changes that cloud systems may undergo at run-time, as well as the complex interactions among multiple software and hardware components, services, and stakeholders. Conformance to data protection requirements in such a dynamic environment cannot any longer be ensured during design time; e.g. due to the dynamic changes imposed by replication and migration of components. It requires run-time data protection mechanisms. This paper proposes combining multiple existing data protection approaches and extending them to run-time, ultimately delivering an end-to-end architecture for run-time data protection in the cloud. We validate the practical applicability of our approach by a commercial case study.**

***Index Terms*—Cloud, Data Protection, Privacy, Architecture**

I. Introduction

Cloud computing offers significant advantages. However, protecting sensitive data, such as personal data or confidential business data, remains a major concern [12]. Users of cloud services lose control of their data by using the cloud, risking that unauthorized parties gain access to sensitive data. Service providers entrusted with handling sensitive data must comply with legislation on data protection (e.g., the General Data Protection Regulation (GDPR) [3]) as well as with individual users' data protection requirements; otherwise, they risk high penalties and reputation damage.

Protecting sensitive data in the cloud is challenging for multiple reasons: 1) *Complexity*: The cloud consists of many different entities – software components like middleware and applications, hardware components, data, stakeholders, business processes etc. [7]. All these entities may need to be protected, and they all may constitute attack surfaces. 2) *Dynamism*: The cloud is continuously changing. New services may be introduced or existing services modified, new users may start using a service or existing users may change their privacy preferences, deployments may change as a result of migrations and replications among servers or data centers, etc. 3) *Conflicting goals*: Data protection is one of multiple goals relating to cloud services. Since data protection mechanisms often impact other goals negatively (e.g. performance overhead), an appropriate trade-off between the conflicting goals has to be found.

Prior work on data protection in the cloud used different security mechanisms, such as encryption [14], secure hardware [6], and access control [13]. While all these approaches are helpful in securing some data transfers or computations, the risk of unauthorized access to data in the cloud still remains high. The deficits of existing approaches stem from multiple sources: (i) individual security measures are not enough to prevent a malicious party from attacking the "weakest link in the chain"; (ii) existing approaches have specific limitations (e.g. require special hardware) or incur considerable overhead (e.g. in the case of fully homomorphic encryption) which make their application impractical in some situations; (iii) during service operation, the environment changes constantly, calling for different security measures at different points in time.

To protect data in a complex and dynamic cloud environment, existing data protection techniques should be combined to an *end-to-end architecture for run-time data protection*. The main characteristics of such an architecture are: 1) *End-to-end*: the architecture must take into account all cloud layers, all relevant stakeholders, and the whole data lifecycle. Protection is necessary throughout these dimensions. 2) *Run-time*: To ensure that the used data protection mechanisms are always in line with stakeholders' current protection needs and the current circumstances and possibilities of the cloud environment, data protection mechanisms should be dynamically activated, deactivated, or reconfigured at run-time.

In this paper, we focus on the design considerations necessary to devise an end-to-end (E2E) architecture for run-time data protection. We analyze the requirements of the relevant stakeholders, present a first conceptual architecture and validate to what extent it fulfills the identified requirements. We use two forms of validation: A scenario-based goal satisfaction analysis and the application in a commercial case study.

An expanded version of this paper is made available in [11].

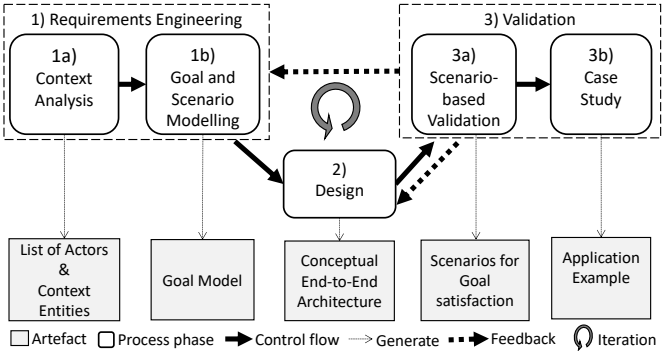

Figure 1. Development and Validation Process of End-to-End Architecture.

II. Process Overview

We followed a systematic and iterative process with three phases, as shown in Figure 1. Phase 1 is concerned with *requirements engineering* [10], including context analysis and goal- and scenario-based requirements identification (1a resp. 1b in Figure 1).

Phase 2 is the actual *design* of the conceptual architecture that addresses the requirements and satisfies the goals. This involves a decomposition into a set of components, definition of the responsibilities of the components, and their interactions.

Phase 3 is the *validation* of the developed E2E architecture. We do this using scenario-based goal satisfaction analysis (step 3a in Figure 1), i.e. using a scenario for each goal of the goal model (regarding data protection) that demonstrates the satisfaction of the goal. Additionally, we instantiate the E2E architecture in a commercial case study (step 3b).

Feedback gathered during the validation phase is fed back to the requirements engineering and design phases as indicated.

The design and validation process was carried out in the framework of the RestAssured project [8], involving 20+ professionals from 6 organizations from academia and industry.

III. Requirements Engineering

To determine the roles and requirements related to data protection, we consider two key formal documents: the privacy framework defined in ISO/IEC 29100 [1] and the GDPR [3].

1a) Context Analysis: We use the following definitions for the key entities and most important actors:

– **Data Subject:** A person about whom data are stored / processed in the cloud. The data subject has, with respect to the personal data about them, the rights stipulated by the GDPR. In this paper, we consider the end-user as a data subject.

– **Data Controller:** A legal entity providing a cloud service which stores/processes personal data. The data controller has the obligations stipulated by the GDPR. We consider the cloud provider at the first interaction point with the end-user as the data controller.

– **Sensitive Data:** Data stored in the cloud that needs to be protected in line with the data protection preferences of the affected data subject. We extend the notion of sensitive data to also include confidential business data.

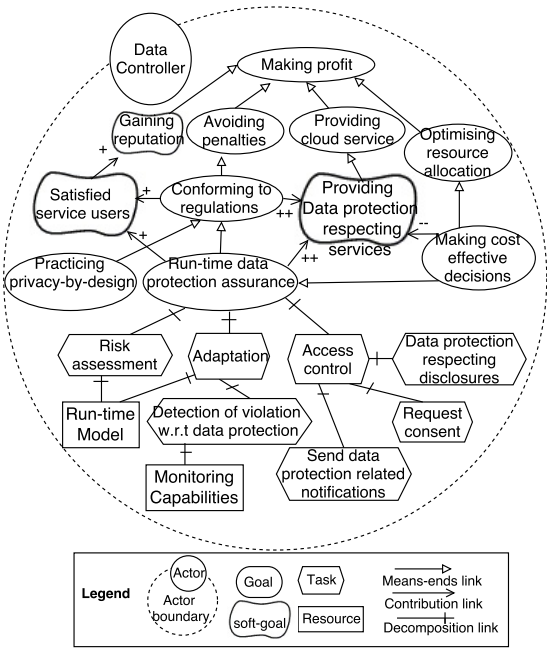

Figure 2. Simplified i* Goal Model including Data Protection Goals.

– **Application:** A cloud application that works with sensitive data. The data controller is responsible for the legal and compliant operation of the application.

– **Infrastructure:** The physical and virtual infrastructure that hosts the application and the sensitive data.

1b) Requirements Identification using Goal and Scenario Modelling: The i* models developed at this stage help in understanding why an E2E run-time data protection system is needed. During the validation phase, the i* models are used to evaluate the developed E2E architecture based on how well it meets the goals and requirements.

In Figure 2, an i* goal model for the data controller is depicted. For the data subject, we set up a similar model, which however is not in the focus of this paper. The overall goal of the data controller is "Making profit". This goal is decomposed into several subgoals.

The data controller is responsible for assuring conformance to data protection regulations such as the GDPR, while providing the cloud services. "Conforming to regulations" can satisfy the goal of "Avoiding penalties" while having positive influences on the soft-goal of "Providing data protection respecting services" as well as having "Satisfied service users".

The data controller can achieve the goal "Conforming to regulations" by "Practicing privacy-by-design" methods and by providing "Run-time data protection assurance". Our focus is on run-time aspects, thus we decompose the "Run-time data protection assurance" goal into three tasks: (i) "Risk assessment" based on "Run-time Model" as resource, (ii) "Adaptation" upon "Detection of violation with respect to data protection", which necessitates "Monitoring Capabilities" for observation of the cloud architecture and (iii) "Access control", requiring to "Send data protection related notifications", "Request consent" and "Data protection respecting disclosures".

To allow for "Optimising resource allocation", "Making cost effective decisions" is necessary, which can have a negative influence on "Providing data protection respecting services". Hence, some conflict resolution needs to be devised.

We derived the following requirements that the E2E architecture needs to satisfy:

- R1 Data subjects should be able to register to the E2E run-time data protection system to specify / update their privacy preferences.
- R2 Accesses to sensitive data of data subjects are only permitted if allowed by the relevant data protection policies.
- R3 Data controllers should be able to register to the E2E run-time data protection system to specify contracts of offered services.
- R4 Applications should be able to request access to sensitive data.
- R5 The application's accesses to sensitive data should always comply with the data protection policies.
- R6 Data controllers should be able to monitor applications, the infrastructure and changes in data protection policies.
- R7 Violations of data protection policies should be identified.
- R8 Data controllers should be supported in performing adaptations on applications and the cloud infrastructure.
- R9 Data controllers should be supported in identifying risks with respect to data protection, thus facilitating proactive adaptations.

IV. Design of End-to-End Architecture

Figure 3 presents an overview of the conceptual E2E architecture for run-time data protection. The dashed frame is the boundary between an application- and infrastructure-agnostic data protection service (the "system") and the set of entities that interact with the run-time data protection system but are beyond the control of our approach (the "context").

For the right interpretation of the overview diagram, the following assumptions have to be taken into consideration:

– The diagram only presents logical components and their logical relations. Deployment considerations (number of instances of each component, co-location or integration of multiple components, centralisation/decentralisation, distribution of the components) are not prescribed by the diagram.

– To simplify the representation, only a single application and a single data controller are shown in the diagram. In practice, multiple applications and data controllers can be handled by the same run-time data protection service instance.

– The diagram only shows relations among components and relations between a component and a context entity. Relations among context entities are not shown.

As depicted in Figure 3, the E2E architecture of the run-time data protection system consists of four functional components (within the dashed frame) and the commonly used run-time model. These components are described as follows:

– The *Run-time Model* is a model of all relevant assets and their relationships within the system and in its context. The model is kept up-to-date using monitoring. The information in the model is used by multiple components to reason about the current situation, the associated risks of data protection violation or other requirement violations. This component addresses requirements R5, R6, R7, R8, and R9.

– The *Data Gatekeeper* manages the data protection policies and service contracts governing the data life-cycle. Data protection policies are specified either by data subjects to capture their individual privacy preferences, or by superordinate

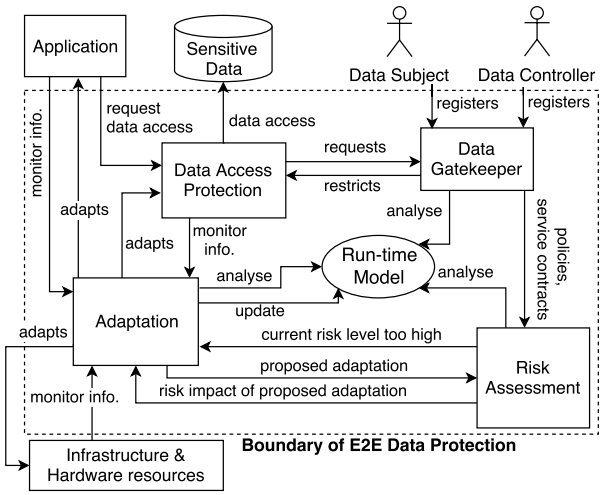

Figure 3. Conceptual End-to-End Architecture for Data Protection.

actors to specify general rules of data protection (legislation, company policies etc.). Service contracts are established with service providers in their role as data controller, defining what operations their service performs on which kinds of data. The *Data Gatekeeper* is responsible for deciding, based on the available policies and contracts, which operations are allowed on which piece of data. This component addresses the requirements R1, R2, and R3.

– The *Data Access Protection* component is responsible for ensuring that data accesses are secure and conform to the relevant policies. To ensure data confidentiality and integrity, the *Data Access Protection* component applies secure enclaves and cryptographic techniques: the data are stored in encrypted form and their decryption takes place in a secure environment, either within a secure hardware enclave or on a trusted machine (e.g. in a private data center). This way, unauthorized parties cannot get access to the cleartext. Moreover, the *Data Gatekeeper* is involved in access control to enforce the compliance with the specified data protection policies. The "Access control task" from the goal model is addressed by the *Data Gatekeeper* and *Data Access Protection* components jointly. The *Data Access Protection* addresses the requirements R2, R4, and R5.

– *Adaptation* is responsible for the satisfaction of requirements in the presence of run-time changes. To this end, the *Adaptation* component continuously monitors the system and its environment. If a change is detected, its impact on data protection and other quality attributes is analyzed. If an actual or imminent problem is identified, *Adaptation* devises a plan to adapt the system such that the problem is avoided or mitigated. Finally, the adaptation is carried out by reconfiguring the appropriate component or context entity. This component addresses the requirements R6, R7, and R8.

– *Risk Assessment* is responsible for continuous run-time assessment of risks. On the one hand, it assesses risks associated with the current system setup, and triggers the *Adaptation* component if the risk level is too high. On the other hand, it assesses the risk impact of planned adaptations to ensure that any changes proposed by adaptation will be compliant

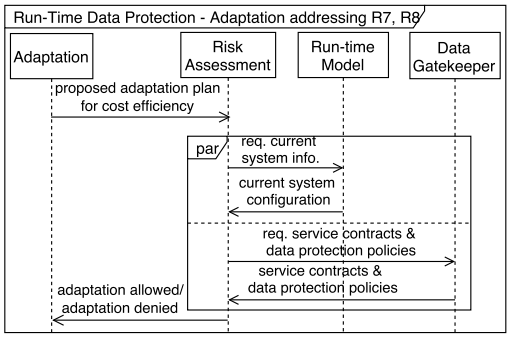

Figure 4. Satisfaction of Run-Time Data Protection Assurance by Adaptation and Resolving Conflict with Cost Effective Decisions.

with the available policies and do not introduce unacceptable risks of data protection violation. This component addresses the requirements R7 and R9.

The interplay of the components and the main data and control flows can be summarized as follows:

– The *Data Gatekeeper* and *Data Access Protection* components cooperate to ensure that an application can only access data that it is allowed to access, based on the data protection policies specified by the individual data subjects. Specifically, the *Data Gatekeeper* combines the query of an application with policy information to form a modified query that is guaranteed to comply with the policies (e.g. by excluding data of data subjects that did not consent to this kind of processing). The *Data Access Protection* component then executes this modified query against the actual database.

– *Adaptation* uses monitoring information to keep the *Run-time Model* up-to-date. The monitoring information is obtained from both context entities and internal components. The *Risk Assessment* component analyses the *Run-time Model*, and if it detects a situation in which the risk of data protection violation is too high, the *Risk Assessment* component triggers the *Adaptation* component. The *Adaptation* component proposes adaptations. If an adaptation is approved by the *Risk Assessment* component, the *Adaptation* component executes the adaptation. For instance, if an application is migrated from a private to a public cloud, *Risk Assessment* would flag an increased risk of data protection violation, which adaptation could mitigate by turning on encryption in the application.

To create an audit trail, all components log their data protection relevant activities to a nonrepudiable logging service.

A fundamental assumption underlying the interfaces between the run-time data protection system and its context is that the context entities trust the run-time data protection system. The trustworthiness of our proposed system is ensured using appropriate technical solutions, e.g. by deploying the critical parts of the architecture on secure hardware. Since the context entities trust the run-time data protection system, the context entities can be expected to provide the necessary interfaces, e.g. for monitoring and adaptation.

V. Validation

We use scenarios to examine the satisfaction of the major goals related to run-time data protection assurance. Figures 4 and 5 depict some example goal satisfaction scenarios as UML sequence diagrams. For further detailed scenarios cf. [11].

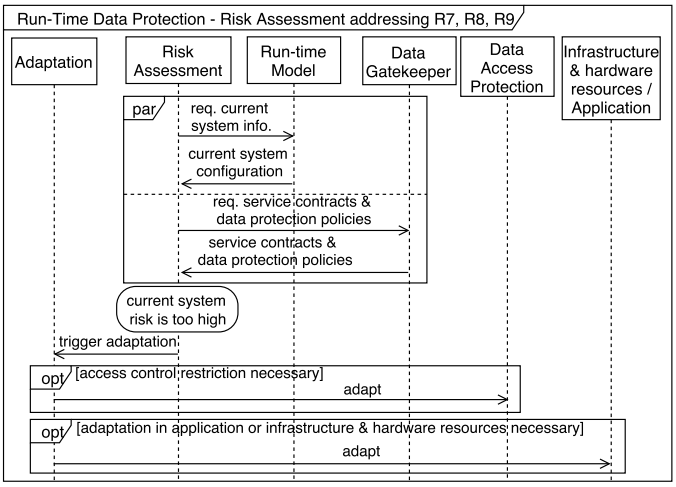

Figure 5. Satisfaction of Run-Time Data Protection Assurance by Risk Assessment.

Figures 4 and 5 show the scenarios satisfying "Run-time data protection assurance" by the tasks "Risk assessment" and "Adaptation", based on the "Run-time Model" as a resource. Figure 4 depicts also the conflict resolution between making cost effective decisions and ensuring data protection (cf. Figure 2). Once the *Adaptation* component comes up with an adaptation plan (e.g. migrations among servers to lower costs), this plan is assessed by the *Risk Assessment* component. *Risk Assessment* uses the current configuration of the system from the run-time model and information about service contracts and data protection policies to evaluate the proposed adaptation plan. Based on the evaluated risk, adaptation can be performed or is denied.

Figure 5 shows how the *Risk Assessment* component can also trigger the *Adaptation* component to adapt the system configuration once it determines that risks of data protection violation are too high in the current system configuration. This may lead for instance to restrictions of data accesses.

Commercial case study*:* we now describe how the proposed architecture is used in a real-world, commercial application that handles personal data in the social care domain.

Ami, developed and operated by Oxford Computer Consultants, is an online service[1] in the United Kingdom that connects (i) lonely people who need help and (ii) volunteers offering help. Matching volunteers to people needing care is based on information such as place where the person lives and their needs. These pieces of information are displayed only in obfuscated form, so as to preserve the users' privacy.

The information about people with loneliness and related needs is valuable to local authorities, who are responsible for supplying social care to persons in need within their areas. *SCANT* is a tool to assist the local authorities in identifying unmet needs, whilst also preserving the privacy of the potentially vulnerable Ami users. For instance, local authorities can

[1]https://www.withami.co.uk/

query with SCANT the number of Ami users with particular needs in a broad geographical region – however, individual Ami users who did not consent to the disclosure of their data must remain anonymous to the local authorities. SCANT is currently being developed according to the approach described in this paper; a first prototype is already available and working. Figure 3 applies to SCANT with the following refinements:

– The registration of data subjects takes place with a special user interface (Ami/SCANT registration tool), which forwards the information about user consent to the data gatekeeper.

– Currently, the SCANT application is neither monitored nor adapted. However, adaptations of the data access protection component indirectly impact the way SCANT can access data.

– For storing sensitive user data, the Opaque secure data analytics platform is used, which provides additional guarantees about data confidentiality and integrity [15].

By using the proposed architecture, data protection policies of the Ami users can be captured and enforced throughout the data lifecycle. The stored sensitive data are protected against unauthorized access. Queries of local authorities are modified automatically on the fly so that the data of Ami users who did not consent to the analytical use of their data are excluded from the results. This guarantees that local authorities *never get access to data of Ami users* who did not consent to this. Local authorities can still work with data of Ami users who did consent to the disclosure of their data as well as with aggregated data of Ami users who consented to this. This automatic fine-grained access control is a major advantage of the architecture. Furthermore, through continuous monitoring, risk assessment, and adaptation, also changes to the underlying infrastructure can be handled transparently. For instance, it is possible to switch between multiple Opaque nodes on the fly provided that they offer similar protection levels, as determined by run-time risk assessment.

VI. Related Work

Various approaches were proposed to address individual aspects of run-time data protection in the cloud. In the following, we review some representative examples. However, so far we lack an integration of these individual approaches into an end-to-end solution for data protection at run time.

Dynamic access control: Veloudis et al. [13] propose an approach for modelling access control policy rules, to support developers in expressing policies for security controls that are appropriate for dynamic and heterogeneous cloud environments. Our work also considers access control (as part of the *Data Access Protection* and *Data Gatekeeper* components), but in combination with other run-time data protection techniques.

Encrypted data sharing: Ibrahim et al. [5] provide a secure data sharing framework using a set of cryptographic techniques. The framework is positioned in the application level targeting confidentiality, integrity, authenticity, availability and auditability. Our work also includes encryption (as part of the *Data Access Protection* component), but combines it with several other techniques for run-time data protection.

Software-based secure processing: Dai et al. [4] provide a trusted execution environment using a software-based trustworthy processing module. The approach only targets the cloud infrastructure level, which is also the case for other hypervisor-based solutions to protect applications, such as [2].

Secure processing using dedicated hardware: Masti et al. [9] provide an architecture for cloud settings where each user receives an independent secure environment. Within a user's independent environment, a user can run sensitive applications. The architecture relies on light-weight processor extensions and hardware-based virtualisation. The approach supports infrastructure providers, but does not consider other data protection requirements beyond the infrastructure level.

Dynamic resource management in the cloud: our previous work [6] bases resource assignment decisions not only on energy efficiency, performance, and cost, but also data protection concerns. Here, we use this work as building block for the *Adaptation* component and integrate it with other data protection techniques realized in the other components of the E2E architecture. In particular, we integrate it with an analysis of the data protection risks implied by an adaptation decision (realized in the *Risk Assessment* component).

Summary. In comparison to existing work, our E2E architecture not only addresses the infrastructure level, but also covers the application level by supporting data controllers in being compliant with data protection policies. The architecture provides an integrated solution for all cloud layers, addressing all relevant stakeholders, and capturing the whole data lifecycle (from design to run time).

Acknowledgment. This work received funding from the EU's Horizon 2020 research and innovation programme under grant agreement 731678 (RestAssured). We gratefully acknowledge constructive discussions with our project partners.

References

[1] ISO/IEC 29100:2011 - Information technology – Security techniques – Privacy framework. 2011.

[2] A. M. Azab, P. Ning, and X. Zhang. SICE: A hardware-level strongly isolated computing environment for x86 multi-core platforms. In *Proc. of the 18th ACM Conf. on CCS*, pages 375–388, 2011.

[3] Council of the European Union. General Data Protection Regulation, 2016.

[4] W. Dai, H. Jin, D. Zou, S. Xu, W. Zheng, and L. Shi. TEE: A virtual DRTM based execution environment for secure cloud-end computing. In *Proc. of the 17th ACM Conf. on CCS*, pages 663–665, 2010.

[5] A. Ibrahim, B. Mahmood, and M. Singhal. A secure framework for sharing electronic health records over clouds. In *IEEE Intl. Conf. on SeGAH*, 2016.

[6] Z. A. Mann and A. Metzger. Optimized cloud deployment of multi-tenant software considering data protection concerns. In *Proc. of the 17th IEEE/ACM Intl. Symp. on Cluster, Cloud and Grid Computing*, pages 609–618. IEEE Press, 2017.

[7] Z. A. Mann, A. Metzger, and S. Schoenen. Towards a run-time model for data protection in the cloud. In *Modellierung 2018*, pages 71–86. Gesellschaft für Informatik e.V., 2018.

[8] Z. A. Mann, E. Salant, M. Surridge, D. Ayed, J. Boyle, M. Heisel, A. Metzger, and P. Mundt. Secure data processing in the cloud. In *Advances in Service-Oriented and Cloud Computing: Workshops of ESOCC 2017*, pages 149–153, 2018.

[9] R. J. Masti, C. Marforio, and S. Capkun. An architecture for concurrent execution of secure environments in clouds. In *Proc. of the ACM Workshop on CCS*, pages 11–22, 2013.

[10] K. Pohl. *Requirements engineering: Fundamentals, principles, and techniques*. Springer, 2010.

[11] RestAssured Consortium. Deliverable D3.2: First high-level architecture and methodology, 2018. https://restassuredh2020.eu/publications/.

[12] S. Schoenen, Z. A. Mann, and A. Metzger. Using risk patterns to identify violations of data protection policies in cloud systems. In *13th International Workshop on Engineering Service-Oriented Applications and Cloud Services*, 2017.

[13] S. Veloudis, I. Paraskakis, and C. Petsos. Foundations for designing, defining, validating and executing access control policies in cloud environments. In *European Conf. on ESOCC*, pages 75–82, 2017.

[14] W. Wang, Y. Hu, L. Chen, X. Huang, and B. Sunar. Exploring the feasibility of fully homomorphic encryption. *IEEE Trans. on Computers*, 64(3):698–706, 2015.

[15] W. Zheng, A. Dave, J. G. Beekman, R. A. Popa, J. E. Gonzalez, and I. Stoica. Opaque: An oblivious and encrypted distributed analytics platform. In *Proc. of the 14th USENIX Conf. on Networked Systems Design and Implementation*, pages 283–298, 2017.