Using Risk Patterns to Identify Violations of Data Protection Policies in Cloud Systems

Stefan Schoenen(✉), Zoltán Ádám Mann(✉), and Andreas Metzger(✉)

Paluno – The Ruhr Institute for Software Technology, University of Duisburg-Essen, Essen, Germany
{Stefan.Schoenen,Zoltan.Mann,Andreas.Metzger}@paluno.uni-due.de

Abstract. Cloud services and cloud infrastructures become increasingly complex and dynamic: many different physical and virtual machines, applications and their components interact and all of these entities may be differently reconfigured, deployed, and migrated during run time. In addition, a multitude of stakeholders may be involved in cloud service offering and usage; e.g., service consumers, cloud providers, data subjects, data controllers, and actual end users. Thus, checking whether cloud services comply with data protection policies when storing or processing sensitive data becomes a challenge due to the involved complexity and dynamicity. We present a model-based approach for identifying violations of data protection policies at run-time. Key elements of our approach are (1) a run-time model to represent the actual cloud system and its stakeholders at runtime, and (2) risk patterns that commonly appear in the context of data protection issues. Our approach aims to find instances of these risk patterns in the run-time model. If an instance of a risk pattern is found, this indicates a risk of data protection violation. We demonstrate the applicability of our approach by using an industry scenario.

Keywords: Cloud computing · Data protection · Privacy
Run-time model · Risk pattern

1 Introduction

The compelling advantages of cloud computing, such as the seemingly infinite resource provisioning without the need for buying costly IT equipment, have made the cloud the platform of choice in many domains [5]. However, using the cloud is often associated with a loss of control since multiple parties – e.g., operators of cloud services – may potentially have access to data and code of applications in the cloud. Thus, storing and processing sensitive data in the cloud poses additional risks, hampering the adoption of the cloud [27].

There are several well-known techniques to ensure confidentiality and integrity of data in a cloud environment, e.g., encryption and authentication

L. Braubach et al. (Eds.): ICSOC Workshops 2017, LNCS 10797, pp. 296–307, 2018.
https://doi.org/10.1007/978-3-319-91764-1_24

technologies. However, they all have some drawback, e.g., in terms of performance overhead (homomorphic encryption) or inconvenience for users (multi-factor authentication). Therefore, deciding which mechanism(s) to use is not trivial and thus a certain knowledge base about when to use which is required.

Traditionally, this decision was made during design by experienced software engineers. However, to support agile deployment processes in a DevOps environment and to leverage the potential of the cloud in terms of flexibility, decisions on data protection mechanisms to use have to be made increasingly at deployment time or at run time. For example, an application may need only light-weight data protection mechanisms if deployed in a private cloud, but full-fledged data protection if deployed in a public cloud. During design time, it may not be clear where the application will be deployed [4,28], so the application must be engineered such that the data protection mechanisms for the public cloud scenario are available but can be turned off if the application ends up being deployed in a private cloud. The mechanisms will then be turned on or off at deployment time. Moreover, the application may be migrated during run time between clouds using live migration [16], in which case data protection mechanisms may need to be turned on or off even during run time.

To enable this flexibility, *decisions on the use of data protection mechanisms must be made automatically* during deployment and run time, of course within bounds set during design time. Thus, the task of software and service engineers changes from making a concrete design decision to *providing the decision logic and determining the information on which the decision should be based*. In particular, this calls for real-time risk identification and assessment that can be used to trigger the appropriate mitigation actions during run time.

In this paper, we focus on the question of *what information is needed* to identify risks of data protection violations. Specifically, we argue that this involves the following pieces of information:

- A model of the – current or planned – configuration of the relevant *assets*, including infrastructure elements, middleware, applications, and data, but also involved actors;
- A set of *risk patterns*, which describe asset configurations that would cause too high risks of data protection violation and hence must be avoided.

The main novelty of our approach is its holistic view in two dimensions: considering (i) all layers of the cloud stack, (ii) from design time to run time. This view is needed to make sound decisions, because data protection is a cross-cutting concern and the required knowledge is established partly at design time but partly only when the system is deployed or adapted [17].

The remainder of the paper is organized as follows: Sect. 2 presents a typical cloud deployment scenario that was created with industry partners to illustrate the potential data protection issues. Sect. 3 gives an overview of our approach for identifying violations of data protection policies. The details are described in Sects. 4, 5 and 6. It is applied to the scenario from Sect. 2 in Sect. 7. Related work is discussed in Sect. 8. The paper is concluded in Sect. 9.

2 A Motivating Example

In this section, we look at an industrial cloud setup and its implications on data protection. This scenario has been devised in the context of the project "RestAssured – Secure Data Processing in the Cloud"[1] with several industry partners. While it abstracts from specific applications, it has been validated to reflect the typical data protection concerns of practical cloud systems.

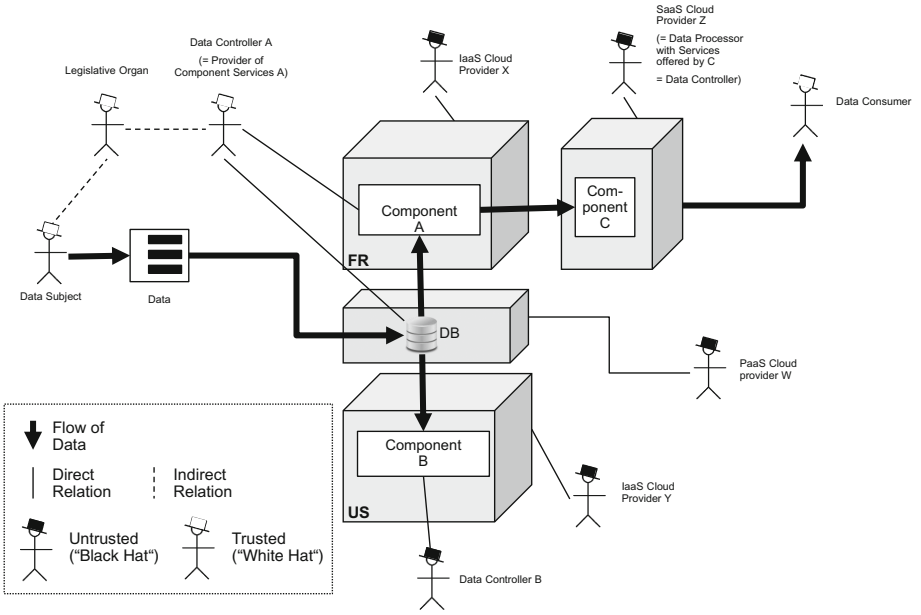

Fig. 1. An industrial cloud scenario

In Fig. 1 we see a typical cloud scenario where multiple parties are involved. In this scenario, personal, unencrypted data (a record) about individual users (Data Subject) are captured by a company (Data Controller A), with explicit consent of the users and under legislative control. The company stores the data in an unencrypted database (DB) operated by a PaaS (Platform as a Service) provider W and deploys its application (Component A) using the infrastructure (including a VM in which Component A runs and a PM in which this VM runs) provided by IaaS (Infrastructure as a Service) provider X. Another actor (Data Consumer) uses an application (Component C) to communicate with Component A and get access to the data. Component C is run by SaaS (Software as a Service) provider Z. Another company (Data Controller B) uses an application (Component B) run on the infrastructure of IaaS provider Y that accesses DB.

[1] https://restassuredh2020.eu/.

It is important to note that Data Controller A and Data Consumer (white hats) are trusted by Data Subject. Nevertheless, as the data traverse between the trusted parties, several untrusted actors (PaaS provider W, IaaS provider X, SaaS provider Z) may get unauthorized access to the data along the way. Moreover, other cloud tenants using the same public database offering (DB) can also get access to the data. The latter also poses a new problem because Component B is hosted in the US, but European regulations prohibit processing personal data of EU citizens outside the EU.

As we can see, a cloud setup can be complex with many different socio-technical interactions, posing a wide-ranging set of threats to data protection.

3 Overview of the Proposed Approach

To address the challenges of ensuring data protection for dynamically deployed and configured cloud services, we devise a model-based approach spanning activities from design time to run time. Figure 2 shows an overview of our approach.

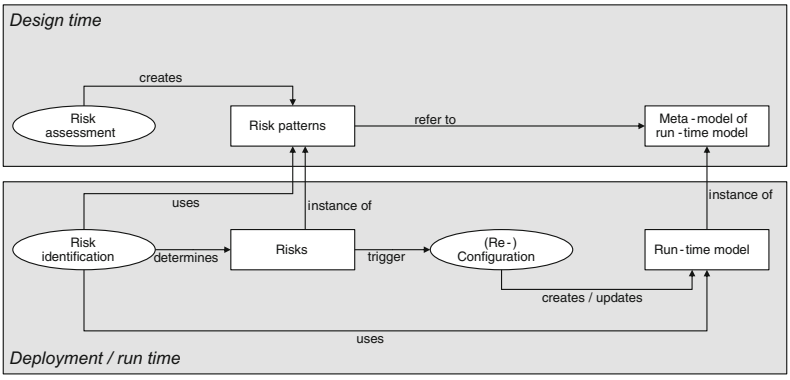

Fig. 2. Overview of our approach. Boxes represent artefacts, ovals represent activities.

Our approach revolves around two types of artefacts: Run-time model and risk patterns. The run-time model (see Sect. 4) is created at deployment time and kept updated during run time, so that it always reflects the current configuration of the cloud-based application and its environment. The risk patterns (see Sect. 5) are created at design time to capture situations that must be avoided because of the associated high risk of data protection violations. These two artefacts are used by the "Risk identification" activity during deployment time and run time to identify risks of data protection violation (see Sect. 6).

4 Run-Time Model

Figure 3 shows an overview of the suggested run-time (meta-)model. It is based on models proposed previously in the literature [10,26], but extends them to have all necessary information for the identification of risks to data protection.

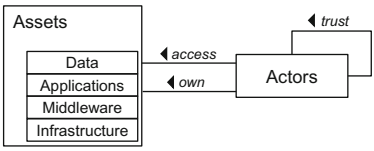

Fig. 3. Overview of the meta-model for run-time models

The run-time model consists of two main parts:

- The *Assets* part allows capturing the current cloud configuration, thereby providing a basis for reasoning about possible data protection violations.
- The *Actors* part allows capturing the relevant actors (natural persons as well as organizations), their roles, attributes, and relations, thereby allowing to reason about their data protection requirements.

These two parts are described in more detail next.

Assets. The most important assets are the data that we want to protect. But other types of assets are also important because they may provide additional attack surfaces (see Sect. 5 for an example). So it is important to consider other asset types as well – here: Applications, Middleware, and Infrastructure.

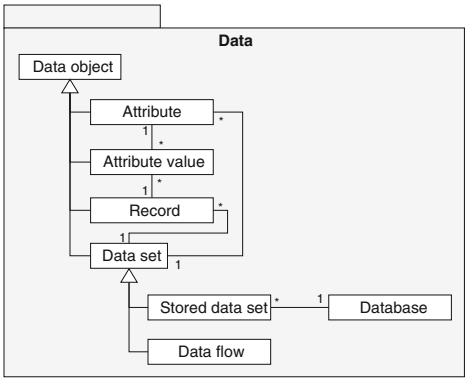

Fig. 4. A possible refinement of the "Data" part of the meta-model for run-time models

The Data, Applications, Middleware, and Infrastructure parts of the run-time model contain the entities that make up the given layer of the cloud stack, along with their attributes and relations. As an example, Fig. 4 shows a possible set of entity types and their relations in the Data part of the meta-model. It should be noted that this is just an example; the exact types may vary depending on the specifics of the used data model (e.g., relational or not).

The Applications part of the model contains information about the structure of the application in terms of components and connectors, the Middleware part contains the entities modeling standard software platform components like database servers and application servers, whereas the Infrastructure part models the underlying physical and virtual resources.

The Assets model must also contain the relations between entities in different parts of the model, for example a data flow that passes through a connector of two application components.

Actors. For modeling the actors, we consider two different kinds of roles:

- Roles related to cloud services. For all involved services (IaaS, PaaS, SaaS), we can differentiate between developers, operators, and users of the services.
- Roles related to data protection. In accordance with the EU's General Data Protection Regulation [7], we use the roles data subject, data controller, and data processor.

Of course, it is possible that multiple roles belong to the same actor. For example, an organization can be user of an IaaS service and operator of an SaaS service.

In terms of the relations among actors, *trust* is of special importance. We use a white-list approach to trust, i.e., every trust relation must be explicitly established (e.g., by means of a contract). Also, trust relations can be limited to specific types of actions on specific data.

5 Risk Patterns

Risk patterns are the core concept in our approach for identifying potential data protection violations. A risk pattern describes a configuration of assets and actors that would lead to unacceptably high risk of data protection violations and hence must be avoided. Syntactically, risk patterns are expressed in terms of the entities that make up the meta-model of the run-time model, their attributes and relations.

Figure 5 shows two examples for risk patterns. In the example of Fig. 5(a), there is a sensitive data record stored in a database operated by a PaaS provider that is not trusted by the data subject. Neither the data record nor the database is encrypted, so the PaaS provider could access the sensitive data, leading to a possible data protection violation. In the example of Fig. 5(b), a personal data record is accessed by an application component that is hosted by a VM in a PM in a non-EU location. Since personal data of EU citizens must not be processed outside the EU, this would also lead to a data protection violation.

To clarify the semantics of risk patterns, it should be noted that the objects in the risk patterns are not specific entities, but should be considered as variables that can take on any specific object of the type as value. The specification of attributes and relations in the risk pattern is to be considered as constraints on these attributes and relations. The statement expressed by a risk pattern is

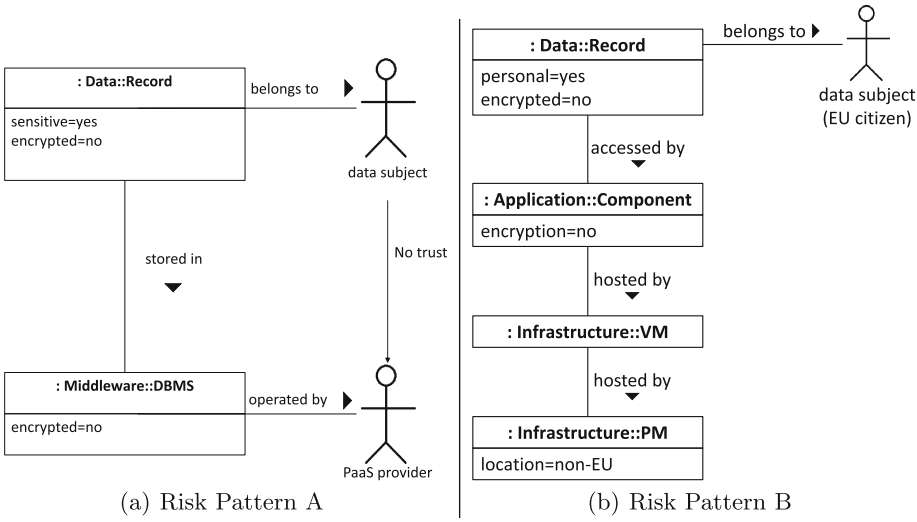

Fig. 5. Two sample risk patterns

that, whenever the variables can be instantiated with specific objects from the run-time model satisfying the given constraints on attributes or relations, this represents an unacceptable case. In the example of Fig. 5(a), the statement can be formulated as follows: it is unacceptable if there is a data subject, a data record, a DBMS and a PaaS provider such that the data record belongs to the data subject, is sensitive and not encrypted and stored in the DBMS, the DBMS is not encrypted and operated by the PaaS provider, and the data subject does not trust the PaaS provider.

The examples show that risk patterns can express complex situations in a compact way. They can adequately capture data protection issues that arise through *interactions of multiple actors and entities on different cloud layers*, also taking into account their attributes and relations.

6 Activities from Design Time to Run Time

Now that we have described the pieces of information forming the core of the suggested approach, we also briefly describe the activities shown in Fig. 2.

Design time. At design time, the meta-model of the run-time model is defined. This also determines the language that is used to express the risk patterns.

Risk patterns are derived from a risk assessment. The risk patterns are designed as a black-list: Each unwanted situation must be captured by a risk pattern. For determining risk patterns, a risk analysis process should be followed, focusing on the types of assets, vulnerabilities and threats in the system. Risks are identified and assessed in terms of their probability and impact. There are well-known methodologies and standards for such a process [12,18].

Deployment time. When the system is deployed in the cloud, the run-time model is created based on the target environment and the planned configuration of the application. When the run-time model is in place, the risk patterns can be evaluated for the first time to check whether the configuration would lead to any data protection issues. The evaluation of the risk patterns can be cast as a graph pattern matching problem, in which one tries to find a subgraph of the run-time model that matches the risk pattern. For this, graph pattern matching algorithms can be used [6,15].

If a match was found, i.e., a risk was identified, the configuration is changed and checked again until a safe configuration is found.

Run time. During run time, the system can use self-adaptation to react to changes in its environment using the MAPE model [14]. Monitoring is used to detect relevant changes and update the run-time model accordingly. For this purpose, existing cloud monitoring tools and further instrumentation can be used. The impact of observed changes is analyzed: the same evaluation logic as during deployment can be used to detect risk pattern matchings in the run-time model. If necessary, the same configuration logic as during deployment can be used to devise a new, safe configuration of the system (planning). Finally, the changes are executed by reconfiguring the system accordingly. The reconfiguration can be done either automatically or using operator-in-the-loop adaptation [13].

It should be noted that adaptation can be also caused by other reason (e.g., insufficient performance). Then, the pattern-matching logic can be used to ensure that the new configuration of the system fulfills data protection requirements.

7 Application to Our Cloud Scenario

In this section we revisit our example cloud scenario from Sect. 2. Figure 6 depicts an excerpt from the corresponding run-time model expressed as an instantiation of the meta-model from Sect. 4. The figure also shows the two matched sample risk patterns from Sect. 5 (framed with dashed lines), as they could be found by a graph pattern matching algorithm:

- Risk Pattern A led to detecting the risk that an untrusted PaaS provider can access a confidential data set stored in the provider's database.
- Risk Pattern B led to detecting the risk associated with the data subject's data being processed on a physical machine outside the EU (here in the USA).

After identifying these specific risks, the configuration of the system can be adapted in such a way that the found risks are mitigated, which will be established by the pattern matching algorithm not being able to find a match with the defined risk patterns. Hence, the run-time model and the risk patterns together can indeed be used to detect violations of data protection policies and – by using the system reconfiguration during runtime – ultimately to avoid them.

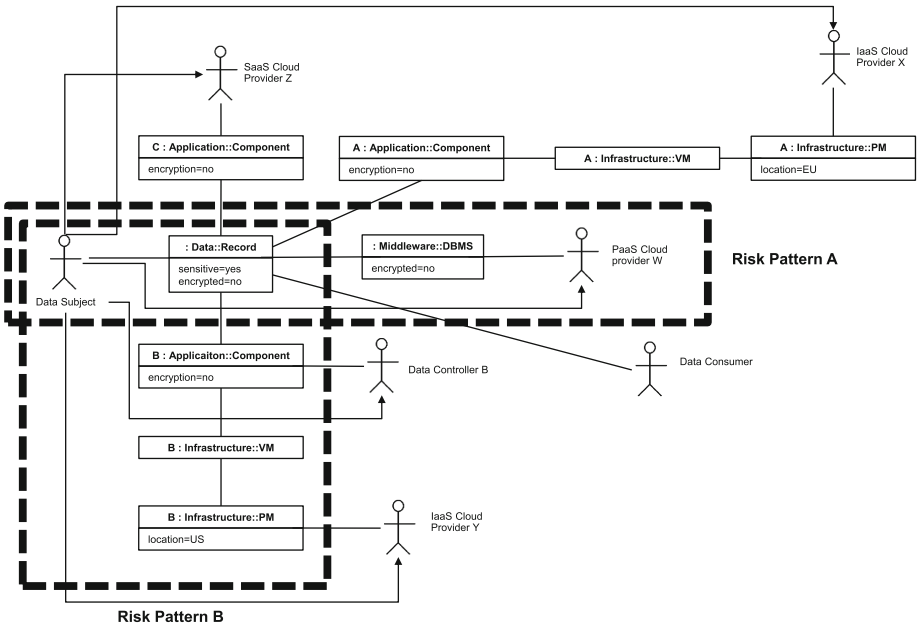

Fig. 6. Run-time model of the example from Sect. 2. The found risk pattern instances are encompassed by the dashed lines.

8 Related Work

We discuss the work most relevant to our proposed approach along the aspects (1) monitoring data protection concerns in the cloud, (2) risk management addressing data protection concerns, and (3) data protection risks of cloud services.

Data Protection Monitoring for Cloud Services. While much work on cloud and service monitoring (e.g., see [1,23]) has been published, approaches for data protection compliance monitoring of services are scarce [20].

Alhamazani et al. [2] discussed the design issues of several monitoring tools and also those of cloud monitoring in general. However, data protection is only mentioned as a minor aspect of monitoring in comparison to performance or QoS regulations. Our approach is mainly focused on data protection.

Foster and Spanoudakis [11] proposed a monitoring approach for cloud services which takes data protection aspects into consideration. Their approach covers only authenticity and auditability as parts of security and ignore other security goals like confidentiality and integrity. Our approach can be used to cover these security goals as well.

Nikolai and Wang propose an IaaS intrusion monitoring and classification system [21]. This system has the purposes of reducing the overload from security sensors by detecting only anomalies which are classified as “urgent” and classifying the attack which causes anormal behaviour. Our approach classifies

these security risks depending if a risk pattern was detected or not. Besides IaaS we also take PaaS and SaaS into consideration.

Watson and Little [28] introduce an approach to reason about the deployment of a distributed system and its impact on security. They state that not all deployment problems can be solved during design time, so run time reasoning is needed. In contrast to our risk patterns, their approach requires the assignment of security levels to all assets, which can be difficult in some settings. Beyond the types of entities considered in that paper, we also explicitly consider actors. As shown in our paper, actors are important for accurately determining data protection concerns.

Meziane et al. [20] describe a monitoring approach for identifying violations with respect to data usage. They introduce privacy-aware SLAs against which data usage flows are checked during run time. Our risk pattern approach takes a broader stance and also covers other aspects relevant for data protection, such as secure storage or processing of data.

In our own previous work, we have introduced a run-time model-based approach for detecting geo-location policy violations [24,25]. Even though this previous work also relies on run-time models, it only considers geo-location as an aspect of data protection, e.g., data may not be migrated outside of the EU. Geo-location is only one relevant aspect for data protection, and so the risk-based approach we introduce here has much broader scope and applicability.

Data Protection Risk Management for Cloud Services. Risk management covers the process of describing, detecting and mitigating risks. So far, only few frameworks for risk management of services have been presented [19].

Djemame et al. [8] propose a risk assessment framework for cloud computing which is designed to help cloud users and cloud providers assess risks during service deployment and operation. This approach focuses on the relationship between service providers and services. However, they do not state how risks may be monitored during operations. This is where risk patterns can help.

Meszaros and Buchalcevova [19] present a framework for online service risk management. They consider similar assets to ours and present a risk and threat model as basis. They focus on risk assessment and mitigation and propose techniques for risk monitoring. Our approach can be considered complementary to their work as we consider the impact of reconfiguration of cloud services on data protection risks.

Data Protection Risks of Cloud Services. Several authors have analyzed specific data protection risks in the context of cloud computing and services. Paquette et al. [22] analyzed the risks of cloud computing, focusing on the context of governmental use of cloud computing. Ardagna et al. [3] survey data-protection-related publications, stating that the areas where risks occur are on application-level, between multiple tenants, or between a provider and a tenant. Fernandes et al. [9] survey security issues in cloud computing as a potential source for data protection risks. These insights provide an important source of input for our approach as they help defining and specifying risk patterns by

taking important data protection concerns into account. Our approach can be seen as a vehicle for capturing and utilizing this kind of knowledge.

9 Conclusion and Future Work

In this paper, we proposed an approach for identifying potential data protection violations in cloud systems. The approach focuses on a run-time model of relevant cloud entities and a set of risk patterns to capture situations that would lead to unacceptably high risks. The identification of potential data protection violations is done using graph pattern matching during deployment time and run time.

The next steps of our research include the formalization of the concept of risk patterns and the adoption of efficient algorithms for the graph pattern matching problem. As a result, we hope to get a better understanding for the possibilities, limitations, and efforts relating to the proposed approach.

Acknowledgments. This work received funding from the European Union's Horizon 2020 research and innovation programme under grant 731678 (RestAssured). Useful discussions with project partners are gratefully acknowledged.

References

1. Aceto, G., Botta, A., de Donato, W., Pescapè, A.: Cloud monitoring: a survey. Comput. Netw. **57**(9), 2093–2115 (2013)
2. Alhamazani, K., Ranjan, R., Mitra, K., Rabhi, F.A., Jayaraman, P.P., Khan, S.U., Guabtni, A., Bhatnagar, V.: An overview of the commercial cloud monitoring tools: research dimensions, design issues, and state-of-the-art. Computing **97**(4), 357–377 (2015)
3. Ardagna, C.A., Asal, R., Damiani, E., Vu, Q.H.: From security to assurance in the cloud: a survey. ACM Comput. Surv. **48**(1), 2:1–2:50 (2015)
4. Brogi, A., et al.: SeaClouds: an open reference architecture for multi-cloud governance. In: Tekinerdogan, B., Zdun, U., Babar, A. (eds.) ECSA 2016. LNCS, vol. 9839, pp. 334–338. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48992-6_25
5. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst. **25**(6), 599–616 (2009)
6. Cheng, J., Yu, J.X., Ding, B., Philip, S.Y., Wang, H.: Fast graph pattern matching. In: IEEE 24th International Conference on Data Engineering, pp. 913–922 (2008)
7. Council of the European Union: General Data Protection Regulation (2016)
8. Djemame, K., Armstrong, D., Guitart, J., Macias, M.: A risk assessment framework for cloud computing. IEEE Trans. Cloud Comput. **4**(3), 265–278 (2016)
9. Fernandes, D.A.B., Soares, L.F.B., Gomes, J.V.P., Freire, M.M., Inácio, P.R.M.: Security issues in cloud environments: a survey. Int. J. Inf. Sec. **13**(2), 113–170 (2014)
10. Ferry, N., Rossini, A., Chauvel, F., Morin, B., Solberg, A.: Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems. In: IEEE 6th International Conference on Cloud Computing, pp. 887–894 (2013)

11. Foster, H., Spanoudakis, G.: Advanced service monitoring configurations with SLA decomposition and selection. In: Proceedings of the 2011 ACM Symposium on Applied Computing (SAC), pp. 1582–1589 (2011)
12. Furuncu, E., Sogukpinar, I.: Scalable risk assessment method for cloud computing using game theory (CCRAM). Comput. Stand. Interfaces **38**, 44–50 (2015)
13. Heinrich, R., Jung, R., Schmieders, E., Metzger, A., Hasselbring, W., Reussner, R., Pohl, K.: Architectural run-time models for operator-in-the-loop adaptation of cloud applications. In: 9th Symposium on the Maintenance and Evolution of Service-Oriented Systems and Cloud-Based Environments (2015)
14. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer **36**(1), 41–50 (2003)
15. Mann, Z.A.: Optimization in Computer Engineering - Theory and Applications. Scientific Research Publishing (2011)
16. Mann, Z.A.: Approximability of virtual machine allocation: much harder than bin packing. In: Proceedings of the 9th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications, pp. 21–30 (2015)
17. Mann, Z.A., Metzger, A.: Optimized cloud deployment of multi-tenant software considering data protection concerns. In: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 609–618 (2017)
18. Martens, B., Teuteberg, F.: Decision-making in cloud computing environments: a cost and risk based approach. Inf. Syst. Front. **14**(4), 871–893 (2012)
19. Meszaros, J., Buchalcevova, A.: Introducing OSSF: a framework for online service cybersecurity risk management. Comput. Secur. **65**, 300–313 (2017)
20. Meziane, H., Benbernou, S., Hacid, M., Malik, Z., Papazoglou, M.P.: A view-based monitoring for usage control in web services. Distrib. Parallel Databases **34**(2), 145–178 (2016)
21. Nikolai, J., Wang, Y.: A streaming intrusion monitoring and classification system for IaaS cloud. In: 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), pp. 632–639, June 2016
22. Paquette, S., Jaeger, P.T., Wilson, S.C.: Identifying the security risks associated with governmental use of cloud computing. Gov. Info. Q. **27**(3), 245–253 (2010)
23. Rabiser, R., Guinea, S., Vierhauser, M., Baresi, L., Grünbacher, P.: A comparison framework for runtime monitoring approaches. J. Syst. Softw. **125**, 309–321 (2017)
24. Schmieders, E., Metzger, A., Pohl, K.: A runtime model approach for data geo-location checks of cloud services. In: Franch, X., Ghose, A.K., Lewis, G.A., Bhiri, S. (eds.) ICSOC 2014. LNCS, vol. 8831, pp. 306–320. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45391-9_21
25. Schmieders, E., Metzger, A., Pohl, K.: Runtime model-based privacy checks of big data cloud services. In: Barros, A., Grigori, D., Narendra, N.C., Dam, H.K. (eds.) ICSOC 2015. LNCS, vol. 9435, pp. 71–86. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48616-0_5
26. Shao, J., Wei, H., Wang, Q., Mei, H.: A runtime model based monitoring approach for cloud. In: 2010 IEEE 3rd international conference on Cloud Computing (CLOUD), pp. 313–320. IEEE (2010)
27. Tang, J., Cui, Y., Li, Q., Ren, K., Liu, J., Buyya, R.: Ensuring security and privacy preservation for cloud data services. ACM Comput. Surv. **49**(1), 1–31 (2016). art. 13
28. Watson, P., Little, M.: Multi-level security for deploying distributed applications on clouds, devices and things. In: IEEE 6th International Conference on Cloud Computing Technology and Science, pp. 380–385 (2014)