2013 IEEE Security and Privacy Workshops

The Cloud Needs Cross-Layer Data Handling Annotations

(Position Paper)

Martin Henze, René Hummen, Klaus Wehrle
Communication and Distributed Systems
RWTH Aachen University, Germany
Email: {henze,hummen,wehrle}@comsys.rwth-aachen.de

***Abstract*—Nowadays, an ever-increasing number of service providers takes advantage of the cloud computing paradigm in order to efficiently offer services to private users, businesses, and governments. However, while cloud computing allows to transparently scale back-end functionality such as computing and storage, the implied distributed sharing of resources has severe implications when sensitive or otherwise privacy-relevant data is concerned. These privacy implications primarily stem from the in-transparency of the involved back-end providers of a cloud-based service and their dedicated data handling processes. Likewise, back-end providers cannot determine the sensitivity of data that is stored or processed in the cloud. Hence, they have no means to obey the underlying privacy regulations and contracts automatically. As the cloud computing paradigm further evolves towards federated cloud environments, the envisioned integration of different cloud platforms adds yet another layer to the existing in-transparencies. In this paper, we discuss initial ideas on how to overcome these existing and dawning data handling in-transparencies and the accompanying privacy concerns. To this end, we propose to annotate data with sensitivity information as it leaves the control boundaries of the data owner and travels through to the cloud environment. This allows to signal privacy properties across the layers of the cloud computing architecture and enables the different stakeholders to react accordingly.**

***Keywords*-Cloud Computing, Data Handling, Privacy**

I. Introduction

Cloud computing offers an abstracted access to a huge pool of resources such as processing, storage, and networking. Instead of having to operate own infrastructure, service providers simply use only the resources they need at a certain point of time, which requires elastic scaling of resources. To receive this elasticity, the resource providers dynamically share resources between customers, which is then called multi-tenancy. Other aspects include a multitude of potentially involved stakeholder (e.g., service and infrastructure providers), the flexible combination of these stakeholders (known as inter-cloud), and location independence. Additionally, the availability of information is increased, e.g., using replication. The huge number of benefits has lead to a wide adoption of the cloud computing paradigm.

In order to identify challenges for data handling in the cloud, we consider one major use case, the handling and storage of all kind of data. Especially when the cloud is integrated with highly sensitive data sources like health-care data [1] or data collected from sensor networks [2], a scaring amount of privacy issues arises [3], [4]. The major concern for users and enterprises is the perception of loss of control over data once it is transferred to the cloud [3]–[7], which has several dimensions. First of all, there is no control over who may access the data, nor any transparency who actually did. Secondly, data might be passed on to third parties or be used for other unintended purposes. Especially for enterprises, it is nearly impossible to guarantee adherence to contracts or laws regarding customer data [5]. Finally, there is no control or at least assurance that data is eventually deleted once it is no longer needed. These concerns are a key barrier to the wide adoption of cloud-based services.

One way to address these privacy issues is security, where one possible measure is encryption. However, simply restricting access to data by means of encryption is not enough to preserve privacy in a cloud environment where data is shared between entities [8]. Encryption, e.g., cannot guarantee that data is deleted after a certain period of time or only stored in certain countries. We argue that data access control (e.g., using encryption) is only one building block for data usage management. It is also necessary to establish trust that data will be handled appropriately. This requires that all entities involved in the handling of data need an awareness how this data has to be treated.

To achieve this, we propose to enrich data in a cloud environment with data handling annotations. Using semantic information for cloud resources has already been proposed to realize federated cloud environments [9]. In contrast, we suggest to extend these ideas to the data being handled in order to address privacy concerns. Our contribution is as follows: First, we present challenges when handling (potentially) sensitive data in a cloud environment. Based on these challenges, we propose an annotation-based approach to data handling in a multi-layered cloud environment. These annotations allow a cloud or service provider to interpret the privacy requirements of the data and handle it accordingly. Finally, we identify and discuss technologies which can be used to realize these annotations in a cloud setting.

II. Data Handling Challenges in the Cloud

Although users and companies could profit a lot from outsourcing data to the cloud, they often refrain from using the

DOI 10.1109/SPW.2013.31

cloud due to privacy concerns [3]–[5]. One major concern is the loss of control over who may access the data once it has been transferred to the cloud. In order to understand this challenges, we first give an introduction to cloud computing. Afterwards we have a look at privacy requirements that lead to challenges when handling data in the cloud.

A. Cloud Computing

The cloud does not consist of one central entity operated by one organization, but involves a number of different stakeholders, distributed all over the world. This holds true especially in a so-called inter-cloud setting, where resources of different clouds are combined [10]. First of all, Infrastructure as a Service (IaaS) providers offer storage and processing resources, which can be rented on demand. On top of these operates the Platform as a Service (PaaS), which abstracts from physical or virtualized resources. At the very top of the cloud stack operates the Software as a Service (SaaS), which targets the end user. The typical end user only interacts with the provider of the SaaS offer she wants to use. This includes that she also only has a contractual agreement with this specific provider and not with the underlying PaaS and IaaS provider(s). However, these have a tremendous impact on fulfilling privacy requirements. In order to answer the question how the user can instruct these providers about how here data should be handled, we first have a look at privacy requirements in a cloud environment.

B. Examples for Privacy Requirements

The cloud paradigm poses a number of challenges to the privacy-aware handling of data. First, the requirements of traditional outsourcing apply to cloud computing as well [3]. Additionally, new requirements arise which are inherent to the cloud paradigm, mainly due to the distributed nature and the desired redundancy.

In the remainder of this section, we will discuss examples for these requirements in more detail. This is not be thought of as a complete list of requirements, but rather as motivating examples for privacy challenges. Additionally, we give high-level ideas, which information needs to be provided in order to be able to address these requirements.

1) Guaranteed Data Deletion: Guaranteed deletion of data is from a user's perspective a key feature of trusted cloud services [3]. From a provider's prospective, the distributed nature and desired redundancy make this a tricky task, especially if reliable deletion methods such as secure data erasure or physical destruction have to be used.

If the storage provider knew in advance at which point in time data should be deleted (e.g., the user requiring deletion after 30 days), it could group data with similar deletion dates on one physical device (replication implies to do this for more than one device). At the right point in time the whole device would then reliably be deleted using secure data erasure or physical destruction.

2) Data Protection Law Enforcement: Certain jurisdictions impose strict data protection regulations when handling personal data. The EU, e.g., demands that personal data of customers must not be transferred to oversea jurisdictions with weaker privacy laws. One prominent exception is known as safe harbor principles, which allow the transfer of personal data to jurisdictions with weaker privacy laws if the recipient declared to voluntarily follow EU regulations.

Nowadays, strictly enforcing data protection laws when using cloud services is nearly impossible. It is nearly impossible to figure out the actual location at which data is stored and there is no way to mark data as data protection law relevant. If the storage provider (at the PaaS level) would know that the data it is currently handling falls under such restrictive jurisdictions, it could evaluate which parts of the infrastructure are compliant to these regulations. The data would then only be store in these parts of the IaaS.

3) Legislative Boundary Awareness: Moving data across legislative boundaries (probably without even noticing), raises severe concerns [3], [4], [11]. This is not limited to data protection, but results from a variety of other legal requirements. One prominent example is the storage of all data relevant for taxes in Germany. This data (and all of its copies) has to be stored in Germany. Only under certain conditions it might also be stored in a different country within the EU or EEA, but never, e.g., in the US.

In order to correctly handle this data, a cloud service would on the one hand need information where this data is allowed to be stored. On the other hand, it needs a way to pass this information to the contracted storage provider(s).

4) Right to be Forgotten: The right to be forgotten is a proposal for a new data protection regulation in the EU [12]. In principal, the right to be forgotten states that personal information has to be deleted automatically after a certain period of time. This addresses the problem that nowadays information which has been released to the internet will never leave it again. Technically implementing the right to be forgotten is considered a challenging task, especially because it stands in stark contrast to US regulations [12].

If the storage provider (IaaS or PaaS) would know whether a data item falls under the EU's right to be forgotten, it could periodically ask the SaaS provider, whether this specific data is still needed and thus trigger the automatic deletion.

III. Cross-Layer Data Handling Annotations

To fulfill the aforementioned requirements when handling data in the cloud, we propose the use of cross-layer data handling annotations. Annotations are a well established method in the field of data usage management [13], [14]. Each entity on the data handling path can add annotations to the data. The other entities than have to treat these as obligations. This is similar to DRM, where access rights are bound to data. More formal, we consider entities in a layered system, where data is exchanged between entities on adjacent layers as well as entities on the same layer. Thereby, we denote the entity that passes data to another entity as *sender* and the one receiving the data as *receiver*. Note, that a receiver might become a sender as well once the data continues traveling. The sender wants to specify obligations regarding how the passed data should be handled. These obligations are then considered binding for *all* receivers on the path. We argue that this approach is better suited than SLAs for fulfilling privacy requirements in the cloud. The dynamic nature of the cloud and constantly changing privacy requirements are difficult to handle with static SLAs.

In the remainder of this section, we will discuss the processes and technologies needed for realizing cross-layer data handling annotations in more detail. For the beginning, we assume that all involved entities are in general interested in benefiting from data handling annotations. Towards the end of this section we will also discuss enforcement of annotations and detection of misbehavior.

A. Annotation Procedure

To illustrate the proposed annotation process (see Figure 1), consider a cloud SaaS service that allows to store and synchronize data with different devices (similar to Dropbox). As motivated in Section II-B1, the user wants her stored data to be ultimately deleted after 30 days. Thus, she annotates the data accordingly before it is handed over to the SaaS. The SaaS checks, whether it can fulfill this obligation and states this to the user. It will then (possibly) choose between different PaaS providers it has under contract and pass the data to one which most likely will be able to fulfill the requirements. Then the PaaS provider will also check, whether it can comply with the obligation and state that fact to the SaaS layer. Again, the PaaS provider hands on the data to a fitting IaaS provider. Finally, the IaaS provider will also check for obligation compliance and report this to the PaaS provider. Then, the IaaS provider has to decide on which part(s) of its infrastructure the data should be stored. As discussed in Section II-B1, it will try to put data with similar deletion dates on the same physical device. Without annotations, this would not be possible.

B. Expressing Annotations

In order to express data handling annotations in a machine-readable way, we propose to use *privacy policy languages* [15]. This is a widely studied field which deals with the formal representation of privacy policies. The formal representation allows to reason about the privacy policies. There are three different types of privacy policy languages: (i) languages that allow users to specify their privacy requirements, (ii) languages that allow service providers to specify their privacy policies, i.e., how they will handle and use data, and (iii) languages that combine the two previous approaches and allow to match or compare a user's requirements against a service provider's policies.

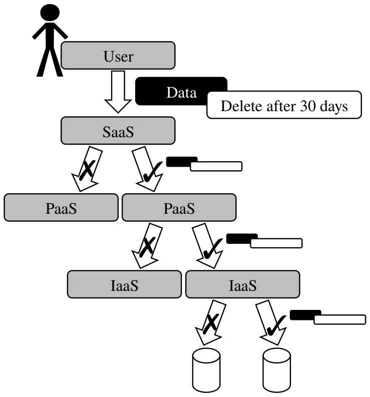

Figure 1. A user adds an annotation to her data ("delete after 30 days") before it is passed to the cloud. Based on this annotation, the SaaS chooses a PaaS, which again chooses a IaaS. The IaaS will then store the data on a physical device together with other data that should be deleted in 30 days.

We argue that in a cloud environment, the third approach is the most promising one, as it allows to formalize the requirements of all involved parties. This would allow the sender to express the data handling obligations and the receiver to formalize the privacy measures it can offer. Thus, when receiving annotated data, the compliance to the stated obligations can be checked automatically. Note that our approach is not bound to a specific privacy policy language.

A number of promising privacy policy languages have been proposed [16], [17]. However, most of these languages are rather technical and require a certain level of abstraction for end users. This could be realized by letting an end user choose between a set of predefined privacy policies. Additionally, these policies could easily be made parametrized, e.g., by choosing the time range after which data should be deleted. The design of some of the languages also allows to delegate (parts of) the policy decision to a trusted third-party [16]. Thus, policies for enforcing, e.g., EU data protection laws, could be retrieved from a central, trusted location.

The formalism introduced by privacy policy languages offers a lot of flexibility [16], but also requires computational effort. However, privacy policies are expected to be rather small and not lead to heavy computations [15]. Furthermore, the same annotation could be used for more than one data item. Instead of sending the full annotation, an identifier for this annotation (e.g., a hash value) would be sent. Thus, we argue that privacy policy languages are well suited for specifying data handling annotations in a cloud environment.

C. Committing to Annotations

In order to establish a chain of trust, we require the receiver of a data item to state its compliance with the annotated obligations. To prevent data to be available without negotiated obligations, the actual data will only be transferred after the receiver has acknowledged its consent. If data would be sent without prior negotiation, an obligation violation could already happen before the obligation is checked. Consider, e.g, the requirement example regarding legislative boundaries (see Section II-B3). Checking for fulfillment of this requirement after the data has already left the country is too late. In order to guarantee the receiver's acknowledgment, we propose a process similar to a three way handshake. As this process requires identities, we assume a public key infrastructure (PKI) to be in place, where (at least) each provider in the cloud stack can be identified by a public/private key pair.

The sender initiates a transmission with an annotation request. It encodes the machine-readable annotation together with a request identifier and sends it to the receiver. In order to establish a linkage between the data and its annotation, we propose to use a hash value of the data as request identifier.

Upon receiving an annotation request, the receiver will parse the machine-readable annotation and decide, whether it can and wants to fulfill the specified obligations. If it cannot or does not want to fulfill the specified obligations, it will send back a negative response. Otherwise, it will reply with an annotation response. In order to confirm its consent, the receiver signs the received annotation request with its private key. The annotation response then consists of the annotation request with the added signature.

Once the sender receives the annotation response, it can verify its authenticity using the digital public key certificate of the receiver. If the authenticity of the receiver's acknowledgment to fulfill the annotated obligations can be verified, it is safe to start the transmission of the data. The sender keeps a copy of the annotation response. In case of misbehavior, it can be used to proof the receiver's consent to the obligations.

D. Binding Data and Annotations

In the previous section we already discussed how annotations can be linked to a data item. Given an annotation, the corresponding data item can thus easily be identified. However, without a way to link data to an annotation, the annotation could be dropped unnoticeable while the data travels through the cloud. Thus, measures to enforce the annotations or detect misbehavior (as discussed below) could not compare the observed conditions to the ones requested.

One approach to binding data to associated policies is the concept of *sticky policies* [18] which got quite some interest in the past years. The underlying concept is to bind a policy cryptographically to the associated data and thus make the policy *stick* to the data. Note that the concept of sticky policies is independent of the representation of policies [19]. Thus, any privacy policy language can be used. Using sticky policies requires the introduction of one or more trusted authorities. Before the sender sends the data to the receiver, it encrypts the data and a hash value of the associated data handling policy. The trust authority's task is to release decryption keys iff it can verify that the receiver states compliance with the policy. Adapting the concept of sticky policies to the cloud has already been proposed [19]. This approach however focuses on which and how cloud services may use data. We see sticky policies as a promising approach to ensure privacy in a cloud environment. It is especially useful when traversing untrusted entities, as the encryption ensures confidentiality.

Another approach for linking data and policies leverages the *integrity protection* mechanism which is often employed for data stored in the cloud [2], [4]. The most common method for ensuring integrity protection of data is the use of digital signatures. For this, a hash value of the data is computed and then signed using public-key cryptography. Anyone in possession of the signee's public key can then verify the signature and thus the integrity of the data. We propose to extend the integrity protection to the annotations associated with the data. This means that the hash value would be computed over the data *and* annotation before it is signed. Thus, unauthorized alteration, deletion, or addition of annotations would break the integrity of the data. Verifying the integrity protection of data in the cloud (including the authenticity of the digital signature) can be efficiently automated using a trusted third-party [20].

E. Policy Enforcement and Misbehavior Detection

In the previous paragraphs we discussed how to annotate data, communicate commitments to obligations, and link data and annotations to each other. Thus, we have created measures for traceability. Still, an open question is how the obligations stated by the annotations can be effectively enforced and misbehavior detected. We now present three complementing approaches that allow to enforce adherence to obligations and detect misbehavior.

1) Auditing and Certification: One established measure to enforce security and privacy in IT systems is auditing and certification. Nowadays, they are highly recommended as a building block to ensure secure data storage, data protection, and policy enforcement in cloud environments [21]. We propose to extend auditing and certification of cloud providers to the verification of the machine-readable privacy policy statements. This would, e.g., include verification of statements on infrastructure location, adherence to data protection laws or the ability to securely delete files.

2) Transparency: Transparency has been identified as a way to establish trust in a cloud provider [11], [22]. On the one hand this refers to disclosing security and privacy mechanisms which are used to protect customers' data. More importantly, it refers to revealing how the actual data of one customer is treated. This could, e.g., mean that a customer could at any point in time look up at which exact physical location her data is stored. Another promising approach to establish transparency are log files [22], which could also state when and how data was securely deleted. Using transparency, users could verify that their annotated obligations are indeed fulfilled. For cloud providers, offering transparency could be an additional selling point.

Partly, transparency can be achieved using auditing and certification (see above). Another possibility is the use of trusted computing which we will discuss in the following.

3) Trusted Computing: Trusted computing (TC) is a technology that ensures (to some degree) that a hardware or software component behaves as expected [23]. Functions enabled by TC include secure input and output, memory curtaining, sealed storage, and remote attestation.

One prominent application of TC is cloud computing [24]. There, trusted computing is, e.g., used to remotely attest the integrity and confidentiality of virtual machines. We propose to use TC to make the policy engine at the receiver a trusted component. Thus, the sender could be sure that the matching of its annotations to the receiver's privacy policies has been performed correctly.

F. Recommendation Systems

Once all the aforementioned mechanisms are in place, one central question still remains unanswered. How to locate and find the SaaS, PaaS, and IaaS provider(s) that are able to fulfill the data handling obligations? At a first glance one might assume that this is a static decision that only has to be made once. However, we believe that this decision process is highly dynamic. The cloud market is always in motion, market players come and go and business models change. Additionally, privacy policies are always in a state of flux. End users might change their perception of privacy, e.g., due to news coverage on data leakage. Cloud providers again might shift their privacy policies based on legislative changes, law suits, or sales reasons. Thus, an approach that is able to identify a fitting provider on demand is essential.

There are already approaches to choose on demand between cloud providers based on the required (technical) resources [10], [25]. These recommendation systems consider Quality of Service (QoS), service-level agreements (SLAs), and pricing as metrics for their decision. We propose to extend these systems to also consider privacy requirements as they are stated in the annotations.

IV. Outlook

We identified challenges when handling sensitive data in a cloud environment. Based on these challenges, we proposed to use cross-layer data handling annotations. With these annotations we are able to communicate obligations regarding the handling of data across the different layers of the cloud stack. We then identified the necessary processes and technologies for such a system and studied them in more detail. All in all, applying data handling annotations to the cloud environment seems a promising approach.

In the future, we plan to further validate the feasibility of our proposed solution. For this purpose, we want to build a prototype of a file storage service (similar to, e.g., Dropbox) able to understand and follow data handling annotations. Additionally, we plan to extend AppScale and OpenStack to support our proposed privacy policy framework.

Acknowledgment

This work has in parts been funded by the German Federal Ministry of Economics and Technology under project funding reference number 01MD11049. The responsibility for the content of this publication lies with the authors.

References

[1] C. Rolim, F. Koch, C. Westphall, J. Werner, A. Fracalossi, and G. Salvador, "A Cloud Computing Solution for Patient's Data Collection in Health Care Institutions," in *Proc. ETELEMED*, 2010.

[2] R. Hummen, M. Henze, D. Catrein, and K. Wehrle, "A Cloud Design for User-controlled Storage and Processing of Sensor Data," in *Proc. IEEE CloudCom*, 2012.

[3] S. Pearson and A. Benameur, "Privacy, Security and Trust Issues Arising from Cloud Computing," in *Proc. IEEE CloudCom*, 2010.

[4] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and Privacy in Cloud Computing: A Survey," in *Proc. SKG*, 2010.

[5] H. Takabi, J. Joshi, and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," *IEEE Security & Privacy*, vol. 8, no. 6, 2010.

[6] I. Ion, N. Sachdeva, P. Kumaraguru, and S. Capkun, "Home is Safer than the Cloud! Privacy Concerns for Consumer Cloud Storage," in *Proc. SOUPS*, 2011.

[7] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses," *Computer*, vol. 45, no. 1, 2012.

[8] M. van Dijk and A. Juels, "On the Impossibility of Cryptography Alone for Privacy-Preserving Cloud Computing," in *Proc. USENIX HotSec*, 2010.

[9] G. Manno, W. Smari, and L. Spalazzi, "FCFA: A Semantic-based Federated Cloud Framework Architecture," in *Proc. HPCS*, 2012.

[10] N. Grozev and R. Buyya, "Inter-Cloud Architectures and Application Brokering: Taxonomy and Survey," *Software: Practice and Experience*, 2012.

[11] J. Heiser and M. Nicolett, "Assessing the Security Risks of Cloud Computing," Gartner, Tech. Rep. G00157782, 2008.

[12] J. Rosen, "The Right to Be Forgotten," *Stanford Law Review Online*, vol. 64, 2012.

[13] A. Schaad and A. Monakva, "Annotating Business Processes with Usage Controls," in *WWW DUMW*, 2012.

[14] A. Aghasaryan, M.-P. Dupont, S. Betgé-Brezetz, and G.-B. Kamga, "Privacy Data Envelops for Moving Privacy-sensitive Data," in *W3C Workshop on Privacy and Data Usage Control*, 2010.

[15] P. Kumaraguru, L. Cranor, J. Lobo, and S. Calo, "A Survey of Privacy Policy Languages," in *SOUPS Workshop on Usable IT Security Management*, 2007.

[16] M. Becker, A. Malkis, and L. Bussard, "A Practical Generic Privacy Language," in *Proc. ICISS*, 2010.

[17] L. Bussard, G. Neven, and F.-S. Preiss, "Downstream Usage Control," in *Proc. IEEE POLICY*, 2010.

[18] S. Pearson and M. Mont, "Sticky Policies: An Approach for Managing Privacy across Multiple Parties," *Computer*, vol. 44, no. 9, 2011.

[19] S. Pearson, M. Mont, L. Chen, and A. Reed, "End-to-End Policy-Based Encryption and Management of Data in the Cloud," in *Proc. IEEE CloudCom*, 2011.

[20] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in *Proc. IEEE INFOCOM*, 2010.

[21] W. Jansen and T. Grance, "Guidelines on Security and Privacy in Public Cloud Computing," NIST Special Publication 800-144, National Institute of Standards and Technology, 2011.

[22] K. Khan and Q. Malluhi, "Establishing Trust in Cloud Computing," *IT Professional*, vol. 12, no. 5, 2010.

[23] C. J. Mitchell, Ed., *Trusted Computing*. IEE, 2005.

[24] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards Trusted Cloud Computing," in *Proc. USENIX HotCloud*, 2009.

[25] P. Pawluk, B. Simmons, M. Smit, M. Litoiu, and S. Mankovski, "Introducing STRATOS: A Cloud Broker Service," in *Proc. IEEE CLOUD*, 2012.