Internal Report 2012–12
October 2012

Universiteit Leiden
Opleiding Informatica

Technology Based Methods to Reduce The Risks of Cloud Computing

Anton den Hoed

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Technology Based Methods to Reduce The Risks of Cloud Computing

Anton den Hoed

Computer Science
Leiden Institute of Advanced Computer Science
Leiden University, The Netherlands

Leiden, October 2012

Master Thesis Project

Thesis submitted in partial fulfillment of the requirements for the degree Master of Science in Computer Science.

Faculty of Science
Leiden Institute of Advanced Computer Science
Leiden University

Supervisors:

First Supervisor	Prof. Dr. Thomas Bäck	(Leiden University)
Second reader	Dr. Hans Le Fever	(Leiden University)
External Supervisor	Rashid Sohrabkhan, M.Sc.	(Accenture)

Author:
Anton den Hoed
Student ID 0639877

Executive Summary

Cloud computing is a computing paradigm that attracts a lot of attention. It promises cost reduction and the outsourcing of IT services and efforts. These advantages cause a rapid growth of the cloud computing market. However, the risks that come with outsourcing software and data to another party are problematic for many companies. Most issues can be traced back to loss of control and regulatory compliance.

The negative effect of the risks on the adoption of cloud computing have led us to conduct a research project with three objectives. First is to find out which risks are most worrisome. Second, to discuss and assess existing technology based methods to reduce four of the most important risks. Third, to introduce new technology based risk reduction methods. While risk reduction can consist of people-, process-, and technology based controls this thesis will focus on technology based methods.

Based on a review of literature on the risks of cloud computing and interviews with cloud computing implementation consultants an overview of the most worrying risks was created. The results indicate that regulatory compliance, data protection, data location, loss of governance and data segregation constitute the top five of most worrisome risks.

In four chapters existing and new methods are discussed and assessed for the following risks:

- Data Location
- Data Deletion
- Data Leakage
- Data Segregation

In the chapter on data deletion a new method is introduced. This method consists of an algorithm that uses data provenance metadata to delete all copies of a data artifact. Because the provenance metadata can be inspected it is possible to verify whether data is really deleted. In the chapter on data location an encryption scheme which is used for digital broadcast is used to constrain which entities can decrypt and use specific data. This method is new to the cloud computing domain.

Other contributions of this thesis are insight into which risks play the biggest role in cloud adoption problems and a (literature) study into existing methods for the reduction of the data location, deletion, leakage and segregation risks.

This study bears some limitations. Because of the scattering of relevant literature and websites and the secrecy of CSPs it is not possible to find all possible sources of existing methods, so the overview of existing methods cannot be exhaustive. Furthermore, the number of experts that were interviewed is only five and they all work for Accenture. While their experience is based on projects with many different customers it is not possible to generalize the results of the interviews.

Contents

Executive Summary 5

Acknowledgements 7

1. Introduction 13

- 1.1. Motivation and Research Goal 13
- 1.2. Research Questions 14
- 1.3. Scope 14
- 1.4. Research Methodology 14
- 1.5. Thesis Outline 15

2. Cloud Computing 17

- 2.1. Delivery models 18
  - 2.1.1. Software as a Service 19
  - 2.1.2. Platform as a Service 19
  - 2.1.3. Infrastructure as a Service 20
- 2.2. Deployment models 20

3. The Risks of Cloud Computing 23

- 3.1. Risk management 23
  - 3.1.1. Risk identification 23
  - 3.1.2. Risk assessment 23
  - 3.1.3. Risk treatment 23
- 3.2. The risks of cloud computing 23
- 3.3. Interviews: The most important risks 25
- 3.4. The risks that will be addressed in this thesis 27
  - 3.4.1. Data Location 27
  - 3.4.2. Data Deletion 27
  - 3.4.3. Data Leakage 27
  - 3.4.4. Data Segregation 27

4. Data Location 29

- 4.1. Introduction 29
- 4.2. Methods to enforce data location 30
  - 4.2.1. Choosing the data location 31
  - 4.2.2. Sticky Policy based methods 31
  - 4.2.3. Data Residency 33
- 4.3. Methods to perform data location monitoring 34
  - 4.3.1. Logging of data location 34
  - 4.3.2. Triangulation of the location of the data 34

4.4. Discussion 35
4.5. Conclusions 36
5. Data Deletion 39
5.1. Introduction 39
5.2. Deletion Methods 39
5.2.1. Disk wiping and physical destruction 39
5.2.2. Crypto-shredding 40
5.3. A data provenance based method for data deletion 40
5.3.1. The Open Provenance Model 41
5.3.2. General idea 42
5.3.3. The algorithm 44
5.3.4. Adapters 46
5.3.5. An example 46
5.3.6. Simulation 47
5.4. Discussion 43
5.5. Conclusions 48
6. Data Leakage 49
6.1. Introduction 49
6.2. Prevention 50
6.2.1. Data Masking 50
6.2.2. Trusted Computing 51
6.2.3. Fog Computing 51
6.3. Monitoring 52
6.3.1. Log file analysis 52
6.3.2. Psychological triggers 52
6.4. Discussion 53
6.4.1. Data Masking 53
6.4.2. Trusted Computing 53
6.4.3. Fog Computing 54
6.4.4. Psychological triggers 54
6.5. Conclusions 54
7. Data segregation 55
7.1. Introduction 55
7.2. Data at Rest 55
7.2.1. Databases 55
7.2.2. File storage 56
7.3. Data in Motion 58

7.3.1. Virtual Local Area Networks 58

7.3.2. Encrypted communication 58

7.4. Data in Use 59

7.5. Discussion 59

7.5.1. Data at Rest 59

7.5.2. Data in Motion 60

7.5.3. Data in Use 60

7.6. Conclusions 61

8. Discussion and Conclusions 62

8.1. Findings 62

8.2. Reflection and limitations 64

8.3. Contribution 65

List of Figures 66

List of Tables 67

List of Abbreviations 68

Appendices 73

Appendix A. Deletion algorithm 73

1. Introduction

In the last few years cloud computing became one of the biggest hypes in the IT world. According to Google Trends the hype peaked in 2011 (Figure 1), but cloud computing is continuing to have a big impact. Analysts from IDC predict that the worldwide spending on cloud services will increase from $17 billion in 2009 to $44 billion in 2013 (Figure 2) [1].

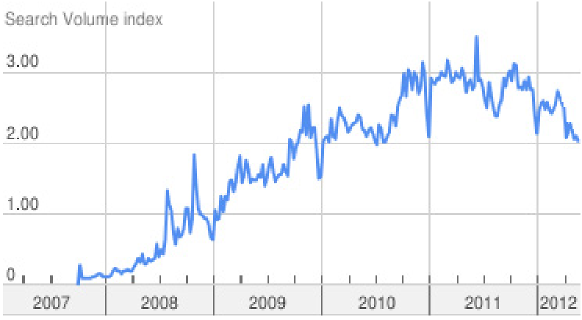


Figure 1. The Google Trends report for the query “cloud computing”.

Moving to the cloud can be complicated because there are many risks which are associated with cloud computing: data location, regulatory compliance, vendor lock-in, data protection; and the list goes on (see [2]). Poor knowledge about the risks and a lack of good methods to mitigate these risks reduces the speed at which companies adopt cloud computing [3].

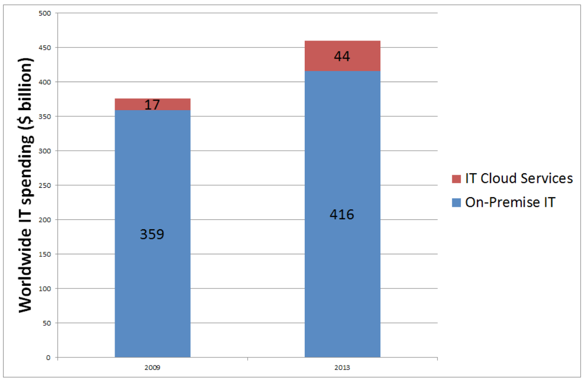


Figure 2. Worldwide IT spending by consumption model (from [1]).

This master thesis aims to find out which data privacy related risks are causing the most concerns and which technology based methods can be used to reduce these risks. These can be methods that are already used in cloud computing, existing methods which can be applied to cloud computing or entirely new methods.

The remainder of this chapter will give a more detailed introduction into this research. Motivation and research goal are discussed in section 1.1. Next, the research questions are introduced in section 1.2. Section 1.3 defines the scope for this project. The research methodology is explained in section 1.4. And last, the outline of the thesis is given in section 1.5.

1.1. Motivation and Research Goal

Moving to the cloud comes with many risks. But how big are these risks? And how can they be mitigated? Companies that want to move (a part of) their IT into the cloud often have these questions, while they can be very hard to answer. Because of this companies are hesitant to use cloud services and the adoption of cloud services is delayed [3].

It can be concluded that the uncertainty about the risks of cloud computing forms a barrier to cloud adoption [3]. This barrier can be lowered by using effective methods to reducing the risks. But which methods are effective to reduce a specific risk? Sometimes it is possible to introduce policies and to make an agreement between the involved parties on which certain rules are enforced. However, in most cases it is necessary to use technology based methods to help enforcing these rules. This thesis will discuss and introduce technology based methods which can help to mitigate the biggest risks of cloud computing.

1.2. Research Questions

To determine on which risks this thesis needs to focus it is necessary to determine which data privacy related risks there are in the cloud computing domain. Therefore the first research question is:

RQ1. What are the (data privacy) risks of cloud computing?

With an overview of the risks it is possible to determine which risks are related to data privacy. It is not possible to cover all the risks, so it is needed to find out how big the risks are and to focus on the biggest risks. Therefore the second research question is:

RQ2. What are the biggest (data privacy) risks of cloud computing?

When RQ2 is answered it is possible to start to answer the third question for each risk that will be covered. To answer this question there are three sub-questions that need to be answered first.

RQ3. What are the possible (and adequate) technologies for reducing this risk?

a. Are there adequate existing technologies which are used to reduce this risk?
b. Are there existing technologies from outside the cloud computing domain which can be translated to a technology which can be used to adequately reduce the risk?
c. If a and b do not yield adequate technologies: Is it feasible to find new solutions? And, what are these solutions?

1.3. Scope

While there are many risks to cloud computing, this thesis will focus on the risks which are related to data privacy. For these risks this thesis discusses, improves or introduces several technology-based methods to reduce them. Because there is more to risk reduction than technology, other methods will also be discussed if they support a technology-based method or if it is not possible to use technology to mitigate the risk.

1.4. Research Methodology

In this section the approach for answering the research questions will be defined.

RQ1 will be answered by performing a literature study. This study includes finding and analyzing relevant literature and creating an overview of the risks.

Based on the risk overview several interviews with cloud computing experts which work at Accenture will be taken. In these interviews the main question is: “Which risks are – in your experience – the risks that raise the biggest concerns for customers?” Based on how many times a risk was identified as an important risk and on the motivations of the expert for choosing certain risks it will be determined which risks are the most important data privacy risks of cloud computing.

For four of the most important risks a study will be done. The study will start with creating a detailed description of the risks and giving examples. After that literature and publicly available information will be used to identify the methods that are available at this moment to mitigate the risk. If the existing repertoire of technology based methods can be improved, research will be done to find methods from outside the cloud computing domain which can be used. Such a method can then be translated to the cloud computing domain and extended to meet the requirements of mitigating a specific risk.

If the previous steps do not yield adequate methods to mitigate the risk it will be tried to create a new technology. When an idea for a new method is found it is worked out in detail, and there will be an attempt to validate the new method.

For each discussed method an analysis of its advantages and disadvantages will be done. It will also be analyzed if the method covers the mitigation of the whole risk, or if there are parts which are not covered and have to be covered in another way.

1.5. Thesis Outline

The remainder of this thesis is organized as follows: Chapter 2 gives a definition of cloud computing and explains its delivery and deployment models. In chapter 3 the risks of cloud computing are discussed, as well as the results of the interviews with experts and an analysis of which risks are the biggest risks.

In chapters 4, 5, 6 and 7 four of the biggest risks are introduced in detail and existing and possibly new risk reduction methods are discussed or introduced.

In chapter 8 the results will be discussed, the research questions will be answered and final conclusions are drawn. This chapter will also contain a reflection upon this research project, an overview of the contribution of this thesis and a discussion of the limitations of this research.

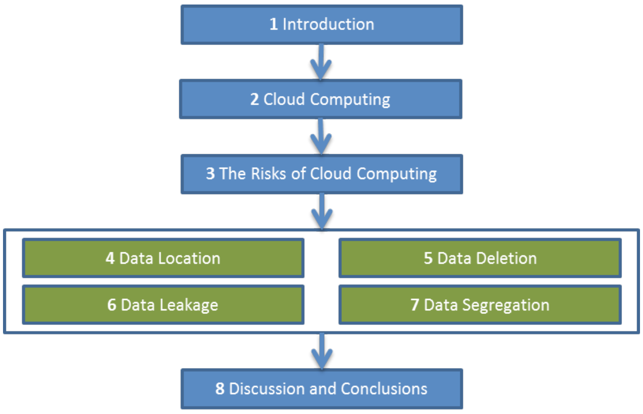


Figure 3. Outline of the thesis.

2. Cloud Computing

From a business perspective, cloud computing is a methodology which enables companies to use IT infrastructure, platforms and software as a service [4]. This means that companies do not need to buy and manage their own hardware and software. With cloud computing a company pays for the services on a pay-per-use basis and almost all management is done by the *cloud service provider* (CSP). Because customers do not need to buy hardware and the applications are delivered via the internet they can vastly reduce their capital expenses. And because CSPs can offer their services in a more efficient way the operational costs for a customer are lower. Another big advantage – besides the outsourcing of IT and the lower prices – is that the time to market can be reduced. Customers subscribe to a service and start configuring and using it right away.

From a technology perspective, cloud computing is the concept of using a large pool of virtualized resources to provide standardized services [5]. All users use resources from the same pool, so the use of hardware is more efficient and users can scale up if they have higher capacity demands. The most important aspects of cloud computing are:

Shared resources

With traditional computing methods each organization has its own hardware. This often leads to underutilization because organizations buy their hardware with peak capacity in mind. Another problem that can occur is that if the demand exceeds the available capacity it is very hard and expensive to scale up.

With cloud computing the resources are shared so customers only have to pay for the resources they actually use. The cloud provider can reduce the amount of hardware because customers do not have the same usage patterns. If one client is in Europe and another in Asia the peak hours for these clients are at different times of the day.

The disadvantage of shared resources is that it is the source of a lot of risks. For example, data from different clients can be stored in the same databases, so it is very important to implement good data segregation.

Standardized services

Whether a CSP provides the possibility to run virtual machines, a development and deployment platform or software services, the services are always standardized. All customers get the same services.

The result is that it is often easy to get a cloud service up and running. Configuration and customization possibilities are well documented.

Pay-as-you-go

Customers only pay for the resources they actually use. This can be per virtual machine, per running instance, per user, per record, or in a number of different ways. Billing can occur on a daily, weekly or yearly basis.

Because customers have no capital expenses for the use of cloud computing, and only pay for the resources they use it is possible to reduce the IT costs.

Delivered via the Internet
Cloud services are often hosted off-premise, so the users access their applications via the internet. Most often this happens via a web browser, a mobile phone or a tablet computer. There is no need to install software and users' devices can be relatively lightweight.

Integration
It is very important that cloud services can be integrated with other applications and services. Therefore cloud providers offer standardized application programming interfaces (APIs) using protocols such as HTTP, SOAP and REST.

2.1. Delivery models

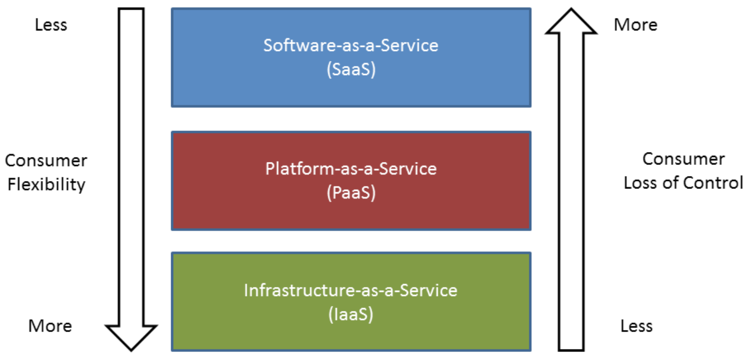
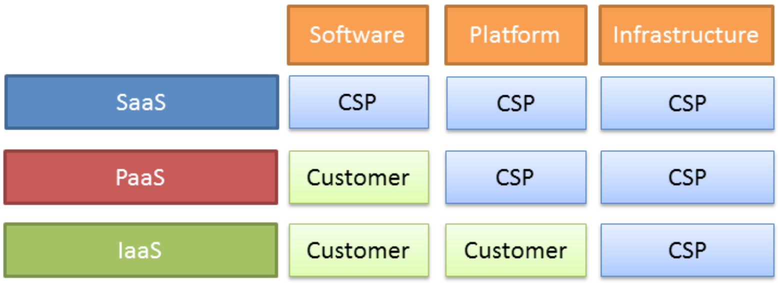

Less
Consumer Flexibility
More
Software-as-a-Service (SaaS)
Platform-as-a-Service (PaaS)
Infrastructure-as-a-Service (IaaS)
More
Consumer Loss of Control
Less

Figure 4. The delivery models of cloud computing.

There are different ways to deliver cloud services (Figure 4). At the lowest level there is the possibility to run virtual machines on the infrastructure of a CSP. This is called *Infrastructure as a Service* (IaaS). One level higher there is the possibility to develop and deploy applications on the infrastructure of a CSP. This is called *Platform as a Service* (PaaS). On the highest level there are standardized applications which are delivered as a service. This is called *Software as a Service* (SaaS).



	Software	Platform	Infrastructure
SaaS	CSP	CSP	CSP
PaaS	Customer	CSP	CSP
IaaS	Customer	Customer	CSP

Figure 5. The responsibilities of the CSP and customer for different delivery models.

Each delivery model has a different division of responsibilities between the CSP and the customer (Figure 5). With SaaS a customer has less control, with PaaS the customer controls the software and with IaaS the customer controls the software and the platform.

The different delivery models will be introduced further in the next three subsections.

2.1.1. Software as a Service

In the traditional situation a company buys licenses for using software which will be installed on hardware which is owned by the company. To receive support and updates it is needed to sign a separate maintenance agreement. This way of working comes with large upfront investments, in-house management and maintenance of hardware and software and underutilization of resources.

With Software as a Service (SaaS) [4][6] a company subscribes to an application which is delivered by a CSP. The subscription includes the usage of the application, support, backups and other services. Because the CSP is responsible for the management and maintenance of the underlying platform and the application, the customer does not have the burden of these responsibilities.

The most important advantages of SaaS are:

- **Outsourcing**: Companies can outsource their software to external providers, reducing their capital expenditure. Because a CSP provides the same service to many customers, they can build a platform at a lower price per customer and thus the operational costs can stay low.
- **Control**: With traditional delivery models the software had to be transferred to customers which reduced the control the vendor had over the usage of their software. With SaaS a vendor has full control over the usage of their software.
- **Economies of scale**: Because a CSP delivers the same service to many customers and it is not needed to provision for peak capacity for each client at the same time the CSP profits from standardization and less hardware requirements.
- **No extra hardware needed**: SaaS applications are delivered via the internet and can be used via a web browser, so a user only needs a computer with a web browser.

Some examples of SaaS applications are Salesforce, Google Apps, Microsoft Office365 and NetSuite.

2.1.2. Platform as a Service

When a company wants to develop a custom application the normal practice is to program the application based on a collection of (standard) components – like database servers and authentication services – which have to be installed and maintained. Deployment of the application is done on hardware which is owned by the company. This is relatively complex and expensive.

Platform as a Service (PaaS) [4][6] is a cloud delivery model which tries to make the development and deployment of applications a less complex and expensive task. A PaaS provider provides a development and a deployment environment to customers. The provider also takes care of automatic scalability, reliability, security and monitoring.

The development environment consists of development tools, standard components and testing facilities. Many PaaS providers provide the following standard components:

- Database
- File Storage
- Queuing Services
- Content Delivery Network
- Large scale data processing

The deployment environment provides the application to the users and automatically scales the amount of resources that are used if there is a high demand for the application.

Some examples of PaaS providers are: Google App Engine, Amazon AWS, Microsoft Azure and Force.com.

2.1.3. Infrastructure as a Service

Infrastructure as a Service (IaaS) is the delivery model where cloud infrastructure is used to deliver a virtualization platform to clients. A client can deploy own or vendor supplied virtual machines to run software in the cloud, and pays for the resources (CPU time, memory, storage and network usage) it consumes. The provider provides a basic virtualization platform with only a limited number of standard services such as backups, scaling, security and virtual machine management.

Some examples of IaaS providers are: Amazon EC2, Cloud.com and GoGrid.

2.2. Deployment models

Deploying a cloud can be done in different ways [6], depending on the requirements of the users of the cloud. A cloud can be open to the public, open to one organization or a specific community. It is also possible to combine different types of clouds.

The first option is a *public cloud*, which is a cloud where every company or consumer can subscribe to the services that are provided by the CSP. The resources of the cloud are shared between the different customers.

When it is not desired or allowed to use a public cloud for specific services it is also possible to use a *private cloud*. With a private cloud the customer controls which services are delivered to which users. There are a couple of different variants of private clouds [4]:

- **Dedicated:** A private cloud which is owned by a single company, and which is hosted on-premise or at a collocation facility. The company has full control over their cloud.

- **Community**: A cloud which is used by a community of companies. It can be owned and hosted by one of the companies or a third party. The community has full control of the usage of the cloud.

- **Managed**: A private cloud which is owned by a single company, but which is managed by a cloud vendor.

If a company uses different services which are hosted in different clouds this is called a *hybrid cloud*. With a hybrid cloud a company can use public cloud based services for non-sensitive and none-core business applications and a private cloud for sensitive and critical applications.

3. The Risks of Cloud Computing

3.1. Risk management

Organizations operate in an uncertain world. Every project or activity has certain risks, but what is risk? According to ISO 31000 risk is the “effect of uncertainty on objectives” [7]. In most cases these effects will be negative, but it is also possible to have positive effects.

Risk can also be defined in a quantifiable way as the probability that a certain event will occur multiplied with the impact that event will have when it happens.

Risk = Probability x Impact

To manage risk effectively it is important (1) to identify and assess the threats to the objectives, (2) to determine the vulnerability of critical assets to these threats, (3) to determine the risk, (4) identify methods to reduce those risks and (5) prioritize the risks.

3.1.1. Risk identification

The identification of threats can be done using many different methods. Examples are brainstorming, workshops, SWOT analysis and scenario analysis. In [8] these and other techniques are discussed in detail.

3.1.2. Risk assessment

To determine how big the risk that a threat poses is, the probability and the impact have to be determined. For some risks it is possible to calculate the impact exactly and determine the probability based on statistical data from comparable activities. However, in many cases this is very hard to determine because there is no hard data to calculate the impact and there is not enough experience with comparable activities to determine the probability. In these cases the impact and probability will have to be estimated by using another method.

3.1.3. Risk treatment

Once a risk is identified and assessed it is possible to determine what can be done to manage the risk. It is possible to avoid the risk entirely, transfer (part of) the risk to another party, accept the risk or to reduce the risk.

Reducing a risk can be done by a combination of people, process and technology. People need to be aware, trained and accountable. Structured and repeatable processes are needed. Technology can be used to continually evaluate the risk and the associated controls, and to monitor and enforce rules which reduce the risk [9].

3.2. The risks of cloud computing

This thesis puts a focus on using technology based methods to reduce the biggest risks of cloud computing. But what are the biggest risks?

The *European Network and Information Security Agency* (ENISA) has performed a risk assessment of cloud computing [2]. Threats were identified by analyzing three different use cases. Experts determined the likelihood and impact of each threat, and the risk levels were estimated by using a risk estimation table which is based on ISO/IEC 27005:2008 ref(see Table 1).

Table 1. Levels of risk which were used in the ENISA study (from [2]).

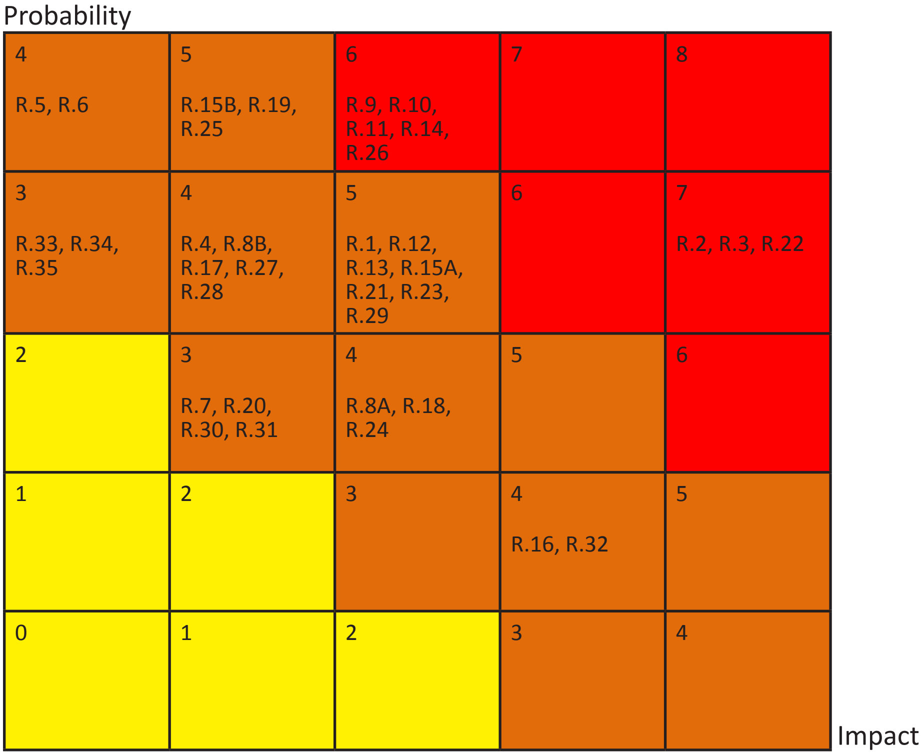
	Likelihood of incident scenario	Very Low	Low	Medium	High	Very High
Business Impact	Very Low	0	1	2	3	4
	Low	1	2	3	4	5
	Medium	2	3	4	5	6
	High	3	4	5	6	7
	Very High	4	5	6	7	8

In total the report identifies 35 risks, divided into four categories. In Table 2 all the risks from [2] are listed by category, and in Table 3 the distribution of these risks by probability and impact is shown.

Table 2. The risks of cloud computing according to ENISA.

Policy and organizational risks		**Legal risks**	
R.1	Lock-in	R.21	Subpoena and e-discovery
R.2	Loss of governance	R.22	Risk from changes of jurisdiction
R.3	Compliance challenges	R.23	Data protection risks
R.4	Loss of business reputation due to co-tenant activities	R.24	Licensing risks
R.5	Cloud service termination or failure		
R.6	Cloud provider acquisition		
R.7	Supply chain failure		
Technical risks		**Risks not specific to the cloud**	
R.8	Resource exhaustion	R.25	Network breaks
R.9	Isolation failure	R.26	Network management
R.10	Cloud provider malicious insider	R.27	Modifying network traffic
R.11	Management interface compromise	R.28	Privilege escalation
R.12	Intercepting data in transit	R.29	Social engineering attacks
R.13	Data leakage on up/download	R.30	Loss or compromise of operational logs
R.14	Insecure or ineffective deletion of data	R.31	Loss or compromise of security logs
R.15	Distributed denial of service	R.32	Backups lost or stolen
R.16	Economic denial of service	R.33	Unauthorized access to premises
R.17	Loss of encryption keys	R.34	Theft of computer equipment
R.18	Undertaking malicious probes or scans	R.35	Natural disasters
R.19	Compromise service engine		
R.20	Conflicts between customer hardening procedures and cloud environment		

Table 3. Risk distribution



Probability

4 R.5, R.6	5 R.15B, R.19, R.25	6 R.9, R.10, R.11, R.14, R.26	7	8
3 R.33, R.34, R.35	4 R.4, R.8B, R.17, R.27, R.28	5 R.1, R.12, R.13, R.15A, R.21, R.23, R.29	6	7 R.2, R.3, R.22
2	3 R.7, R.20, R.30, R.31	4 R.8A, R.18, R.24	5	6
1	2	3	4 R.16, R.32	5
0	1	2	3	4

Impact

ENISA classifies loss of governance, vendor lock-in, isolation failure, compliance risks, management interface compromise, data protection, insecure or incomplete data deletion and malicious insiders as the top risks to cloud computing.

In "Top Threats to Cloud Computing" [10] the Cloud Security Alliance identifies the seven biggest risks to cloud computing: Abuse and nefarious use of cloud computing, insecure interfaces and APIs, malicious insiders, shared technology issues, data loss or leakage, account or service hijacking and an unknown risk profile.

Gartner identifier the top risks in "Assessing the Security Risks of Cloud Computing" [11]: Privileged user access, regulatory compliance, data location, data segregation, recovery, investigative support, long-term viability and availability.

Table 4 shows an overview of the top risks which are identified by ENISA, CSA and Gartner.

3.3. Interviews: The most important risks

The risk analysis documents of ENISA, CSA and Gartner have resulted in a list of the most important risks of cloud computing. However, it is not possible to cover all these risks in this thesis. To determine which risks have the highest priority five cloud computing experts from Accenture were interviewed.

Before an interview the expert received a document with an overview of the top risks from the three different analyses (Table 4) and instructions on how to access the risk analysis documents. During the interview the following question was asked: "Which risks are – based on your experience – the most important risks?". Interviewees were given the possibility to introduce risks which were not on the list.

They were also asked to motivate their choices.

Table 4. The top risks of cloud computing according to ENISA, CSA and Gartner.

ENISA [2]	**CSA** [10]	**Gartner** [11]
Loss of governance	Abuse and nefarious use of cloud computing	Privileged user access
Vendor Lock-In	Insecure interfaces and APIs	Regulatory compliance
Isolation failure	Malicious insiders	Data location
Compliance risks	Shared technology issues	Data segregation
Management interface compromise	Data loss or leakage	Recovery
Data protection	Account or service hijacking	Investigative support
Insecure or incomplete data deletion	Unknown risk profile	Long-term viability
Malicious insider		Availability

During the interviews no new risks were introduced to the list. In Table 4. it is shown how many times a risk was indicated as one of the most important risks during the interviews.

Table 5. The risks that were mentioned as being very important and the number of times they were mentioned.

Risk	**#**
Regulatory compliance	5
Data protection	4
Data location	4
Loss of governance	3
Data segregation	2
Availability	2
Vendor lock-in	1
Insecure or incomplete data deletion	1
Insecure interfaces and APIs	1
Privileged user access	1

Every interviewee classified regulatory compliance as one of the top risks of cloud computing. A company needs to comply with a plethora of regulations and laws, such as privacy laws, accountability laws, data retention rules and internal data protection policies.

Data protection was also mentioned in almost every interview. Most applications contain (privacy) sensitive data that should not be accessible to or modifiable by third parties. However, it is not immediately clear which security measures are implemented by a CSP. Most CSPs also do not offer the possibility to perform specific audits. Customers have to rely on standard reports.

A risk which is a result of regulatory compliance and data protection is data location. In the European Union the privacy laws state that personal identifiable information from EU citizens may only be stored within the EU or in countries with an adequate level of privacy protection [12].

With outsourcing IT activities a company also outsources a part of the controls they have, so loss of governance is also a hot topic. Software and services will be running on hardware that is managed by a CSP and security and privacy controls are the essentially the same for each customer. The rise of Bring Your Own Device (BYOD) also makes the consumption of (cloud) IT services less controllable.

Data segregation can be seen as being part of data protection. The multitenant architecture of cloud computing implies that the data of different customers is stored in a single environment. This necessitates effective segregation technologies and controls in the data storage, transport and processing stages.

Outsourcing IT to another location requires a cloud which is always available. Many customers are afraid that the cloud or their internet connection will encounter outages and work will come to a halt.

Vendor lock-in, insecure or incomplete data deletion, insecure interfaces and APIs and privileged user access were each mentioned once during the interviews. The arguments that were motivating the choices for these risks are mainly the same as the arguments that were used for the other risks.

3.4. The risks that will be addressed in this thesis

This thesis discusses existing – and introduces new – technology based risk reduction methods for four of the most important risks. Because the combination of people, process and technology based controls which can be used to reduce each risk is vastly different it is not possible to cover all types of risk in this thesis. Two of the most important risks – regulatory compliance and loss of governance – will not be covered directly in the following chapters. However, many other risks are a result of these risks, so they will be covered indirectly.

3.4.1. Data Location

The EU privacy directive [12] demands that privacy sensitive data is stored within the EU or in countries which effectively protect the privacy of EU citizens. This means that personal data from the EU may not be stored in most countries. Most CSPs do not fully disclose where data is stored. Even if a CSP gives their customers means to control data location there are exceptions in the terms of service which grant the CSP the possibility to move personal data if necessary.

3.4.2. Data Deletion

Cloud services make many copies of the same data. File systems are replicated, backups are made, etc. This makes it very hard to verify if data deletion is effective. Will all copies be deleted? And, is the deletion process irreversible?

3.4.3. Data Leakage

Intentional or unintentional data leakage by insiders is a problem which is faced by many companies. To mitigate this risk many controls – such as endpoint protection and encryption – are used. However, with cloud computing more and different devices connect to the cloud. Installing effective monitoring software on all devices is not always possible.

3.4.4. Data Segregation

Problems with data segregation can be seen as a more specific instance of data leakage. Effectively separating data of different customers is very important. The well-known controls in this field focus on the separate storage of data; however data will also be transported and processed. Which controls are available to separate data while it is transferred over the network or used by a processor?

4. Data Location

4.1. Introduction

When a company plans to migrate an application to the cloud it usually involves moving data from the premises to a data center of the CSP. But where is that data center? Finding the answer to this question is not as easy as finding out where the CSP is registered, because they can deliver their services from any data center or even multiple data centers.

Existing literature on the risks of cloud computing often classifies data location as an important risk [2] [11]. In the remainder of this section the risk aspects of data location in the cloud are discussed. Because data location is closely related to privacy regulations there will be a focus on personal data.

If a company wants to store personally identifiable information (PII) in the cloud several problems might arise:

- Is it allowed to move PII to the country where the data will be stored by the CSP?
- Can authorities access the data if they need it for legal purposes?
- Is it possible to get information about where the data is stored in case the CSP has data centers in multiple countries?
- Is it possible to enforce a data location policy?

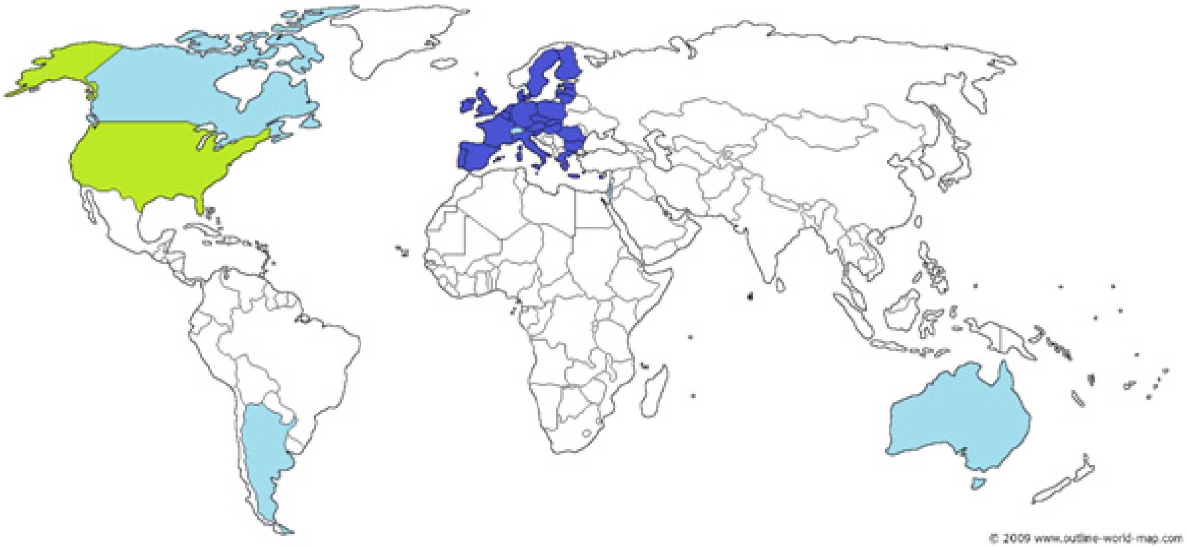
Transferring personal data to another country can be difficult. The European Union demands from their member countries that they implement privacy laws which are based on the Data Protection Directive (Directive 95/46/EC [12]). The directive allows the transfer of personal data between EU member states, but it prohibits the transfer of personal data to a third country if that country does not ensure an adequate level of protection. The European Commission decides which countries have an adequate level of protection. At this moment Andorra, Argentina, Australia, Canada, Switzerland, Faeroe Islands, Guernsey, State of Israel, Isle of Man and Jersey are fully recognized as countries with an adequate level of protection (Figure 6) [13]. The transfer of personal data to the United States is only allowed if the recipient of the data has a certificate which confirms that it adheres to the US Department of Commerce's Safe Harbor Privacy Principles or if the transfer concerns air passenger name records [13].

For a company which is based in an EU member state the national privacy laws apply to the collection of personal data. Moving the data to another country does not change that. The company is also liable for the handling and protection of their data if they place their data in the cloud. This implies that a company has to ensure that their CSPs act according to the national privacy laws.

Access to data by governments is always a big topic in discussions on data privacy ([10][11]). In most countries the government of the country where a CSP is located can issue warrants to retrieve data. In cases like anti-terrorism it is sometimes the case (for example in the United States [16]) that a CSP is not allowed to disclose to their customers that they have received a warrant for the disclosure of their data.

If a CSP has a subsidiary in a country or does systematic business in that country, the CSP has to comply with the local laws. Thus, if a CSP offers their services to customers in the US they have to comply with US law. Even if a CSP has no connection with a country it is possible that the government of that country gets access to data which is stored by that CSP, if a third country has a *Mutual Legal Assistance Treaty* (MLAT) with the country where the CSP is located. The United States has MLATs with over 60 countries [17].

Figure 6. Countries with an adequate level of protection. EU member states are dark blue, fully recognized countries are light blue and the US is green.



© 2009 www.outline-world-map.com

Knowing where your data resides can be very difficult in the cloud. Many large CSPs have data centers in different countries, and they often use distributed file systems which automatically copy data to different data centers to ensure data availability if one of data centers goes offline. Because the algorithms which decide to which data center data is copied are not disclosed, it can be completely unclear where your data is stored. Some CSPs – like Microsoft – address this problem by giving their customers the possibility to choose where their data is stored [18]. Other CSPs – like Google – do not give their customers the possibility to choose where their data is stored [19].

Enforcing the location of data is even harder than knowing the location of the data. Even if CSPs let their customers choose where their data will be stored, there are exceptions in the terms of service which allow moving the data to other places: for solving problems, support or to comply with requests from law enforcement agencies.

4.2. Methods to enforce data location

The simplest method to mitigate the risk of data location is choosing a CSP which only has data centers in specific countries.

If the CSP has data centers in multiple countries and is not willing to guarantee that data is stored in specific countries, it is possible to encrypt the data. If a CSP only stores the encrypted data but does not have the private key they do not store the actual data. This also holds for PII: according to EU member states' privacy laws encrypted PII is no PII if an entity is not able to decrypt the data in any feasible way [20].

A very important problem with encryption of data without giving the CSP access to the private key is that it is impossible for the cloud service to perform processing tasks on the data. Processing data is very important in almost all applications: report generation, automated workflows, search engine indexing, etc.

4.2.1. Choosing the data location

A CSP may offer their customers the choice of where their data should be stored. Microsoft Windows Azure uses this method by letting their customers choose if their data will be stored in the United States, Europe or Asia [18].

4.2.2. Sticky Policy based methods

In [21] Karjoth et al. introduce the *Platform for Enterprise Privacy Practices* (E-P3P), which is a method that enables enterprises to define formalized privacy policies and to enforce these policies. When a data subject enters data he can agree to the applicable privacy policy as well as selecting opt-in and opt-out options for specific parts of the policy. For example, a person wants to create an account at an e-commerce website. He indicates that he agrees with the privacy policy and he opts out of the option to receive a weekly news mailing.

E-P3P attaches the consent information to each data record or file (sticky policies), so it is possible to control data on a very fine-grained level. Major advantages are the possibility to discriminate between different versions of a privacy policy or to have different policies for data subjects which are living in different countries. Within the scope of data location this leads to the idea that a data subject or enterprise can use a sticky policy to define where data may be stored or used.

Karjoth et al. [21] state that their methodology “protects personal data within an enterprise with trusted systems and administrators against misuse or unauthorized disclosure”. With public cloud computing personal data is transferred to another enterprise where the systems and administrators are not controlled by the enterprise.

To counter this problem it is possible to use encryption techniques [22] where a data subject encrypts his personal data with a general public key and attaches a sticky policy. These techniques ensure that only the private keys of approved data users are able to decrypt the personal data.

In the next subsections two cryptographic methods which use this concept are discussed.

4.2.2.1. Broadcast encryption

Broadcast encryption [23] is an encryption method which supports the specification of who can decrypt data. Each entity which should be able to decrypt documents receives a unique private key. A data subject can explicitly define which private keys are able to decrypt the data. An implementation consists of the following three methods:

`Setup(n)`

This method generates n private keys $d_1, \ldots, d_n$ and a public key `PK`.

`Encrypt(S,PK)`

The data subject defines which keys can decrypt the data by creating a set `S` which contains the identification numbers (`1,…,n`) of the selected private keys. Encrypt takes this set and the public key as parameters and produces a header (`Hdr`) and an encryption key (`K`) for the data.

Now the data (`M`) can be encrypted using a symmetric encryption method: `CM = SymCrypt(M, K)`. The result of the Encrypt method is `(S, Hdr, CM)`.

`Decrypt(S, i, di, Hdr, PK)`

Decryption of the data starts with checking if `i` is in the set `S` (which is taken from the encrypted data container `(S, Hdr, CM)`). If `i` is in `S` and d_i is the valid private key of entity `i`, `Decrypt` returns the symmetric encryption key `K`. The original data can then be obtained by executing `SymCrypt(CM, K)`.

In the context of data location this method can be employed by generating a large number of private keys, for each user and each region. Setting the number of keys to – for example – 2^{32} will result in a collection of over four billion keys. Each user receives the global public key and a private key. Each server in the data centers of the CSP receives the private key which is associated with the region where the server resides.

A data subject starts by defining a set which contains the set of allowed region identification numbers. Because the data subject should be able to decrypt his own data his own identification number is also added to the set. With the set and the public key the `Encrypt` methods generates the key `K`, which is used to encrypt the data. The result of this step is stored in the database or file system of the CSP.

When a server needs to access the data it retrieves the encrypted data, the header and the set `S`. Next, the server performs the decryption operations, which only succeed if the server is in an approved region.

The example in Figure 7 demonstrates how broadcast encryption works.

4.2.2.2. Ciphertext Policy – Attribute Based Encryption

With *Ciphertext Policy – Attribute Based Encryption* (CP-ABE) [24] the data subject is able to encrypt data using a global public key and a constraint on the values of different attributes. Each entity that has to decrypt data has a private key that is generated by giving values for the different attributes that are defined. CP-ABE ensures that the data can only be decrypted if the constraint is satisfied by the attribute values of the entity.

To apply this method to data location, each server in the data centers of the CSP has a private key in which a region attribute is set to the region where the server resides. Special care must be taken to ensure that the private keys are not transferred to other machines after they are generated by a key distribution service within the same region.

```
S = {1, 2};
      (Hdr, K) = Encrypt(S, PK);
      CM = SymCrypt(M, K);
      Store (S, Hdr, CM).

Retrieve (S, Hdr, CM);
      K = Decrypt(S, 2, d2, Hdr, PK);
      M = SymCrypt(CM, K).

Retrieve (S, Hdr, CM);
      K  = Decrypt(S, 3, d3, Hdr, PK)
      -> Fails, because 3 is not in S.
```

Figure 7. In (1) the data subject (*id = 1*) defines that he and a region with *id = 2* may decrypt the data. In (2) a server in the region with *id = 2* successfully decrypts the data, while in (3) the decryption fails because the server is in the region with *id = 3*.

The method works as follows: The data subject defines in which regions his data may be decrypted by setting the constraint to – for example – “`region = Europe`”. After this `ci = Encrypt(pk, data, constraint)` is generated and sent to the cloud application.

When a server in Europe needs the data of the data subject, the server retrieves the encrypted data and tries to decrypt it. If the constraint evaluates to true using the attribute values from the server’s private key the original data is obtained. Otherwise the decrypt function returns `false`.

The following example illustrates how CP-ABE works when it is used for file based encryption:

```
cpabe-keygen –o EUserver1_priv_key pub_key master_key \
      ‘region = Europe’
cpabe-keygen –o USAserver1_prev_key pub_key master_key \
      ‘region = USA’

cpabe-enc pub_key document.pdf ‘region = Europe’

cpabe-dec EUServer1_priv_key document.pdf.enc
      -> Document.pdf
cpabe-dec USAServer1_priv_key document.pdf.enc
      -> False
```

Figure 8. Usage of CP-ABE for file encryption. In (1) a private key is generated for a server in Europe, and in (2) for a server in the USA. The data subject defines the region and encrypts his data in (3). In (4) the European server tries to decrypt the file and succeeds, while the USA based server cannot decrypt the file (5).

4.2.3. Data Residency

Another data protection method is data residency [25]. With data residency sensitive data will be tokenized or encrypted by an on premise proxy server before it is stored in the cloud. This means that sensitive data does not leave the premises of the customer, while it is still possible to benefit from cloud services. A CSP that offers this method is Salesforce [25].

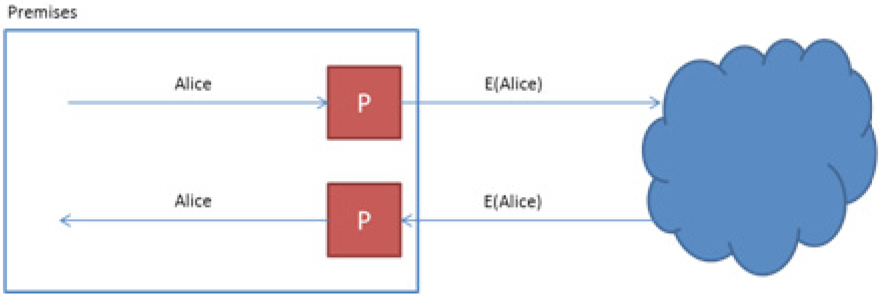


Figure 9. The string 'Alice' passes through an encryption proxy before it is transferred over the internet to a cloud environment. When the data is retrieved the cloud responds with the encrypted data and the on premise proxy decrypts the data again.

4.3. Methods to perform data location monitoring

In this section two methods for monitoring the location of data are described. The first method concerns the logging of trans border movements of data by the CSP, while the second method allows a customer of a CSP to estimate where the data center which contains his data is located.

4.3.1. Logging of data location

A CSP can implement methods which give their customers insight into the location of their data. This can be a dashboard which reports where data is being stored. Logging of data location for such a dashboard can be done using two different approaches.

The first approach is to log the data location at the moment when it is stored. In this way it is possible to report in which countries the data is stored on the file systems of the CSP. But what if the data is copied to another storage medium which does not support the logging methods? In general it is not possible to guarantee that all copy operations on data are logged.

The second approach can provide more insight into the location of copies of data by logging all data movements to other countries. If a user accesses a record via a browser from another country data is moved to the country where the user resides. This implies that all reading operations on the data should be logged.

4.3.2. Triangulation of the location of data

In [26] Peterson et al. propose a method which uses internet based triangulation to monitor data location. The idea is to retrieve data from multiple locations which are distributed around the globe. Because data cannot travel faster than light there is a lower bound on the response time of the requests. Furthermore, network equipment through which the data flows also incurs an extra delay. When a request is repeated the response time will not be constant.

By using the response time and information about the network topology it is possible to find the approximate distance between the probing nodes and the server which responds to the request. Because of the variance in the delays it will be necessary to repeat the measurement several times to get an accurate approximation of the distance between the probing nodes and the server.

With the resulting distances it is possible to triangulate the location of the server with quite high accuracy. An overview of this method is depicted in Figure 10.

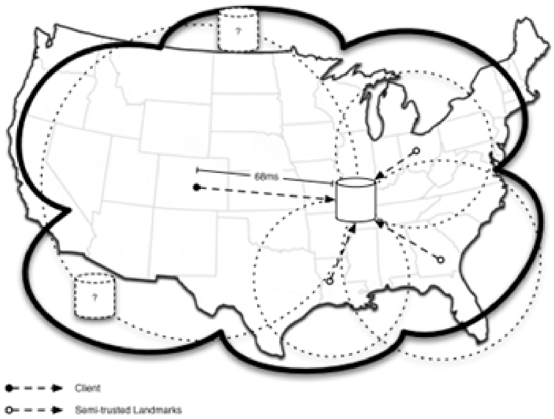


Figure 10. Finding the location of a server by triangulation. The dots represent the probing nodes, the dotted circles represent the distance between the probing node and the server. The place where the circles intersect is the probable location of the server. (Figure from [26])

4.4. Discussion

In this chapter several methods for enforcing and monitoring data location in the cloud were covered. In this discussion an analysis will be made of the adequateness of the methods by looking at their characteristics, advantages and disadvantages.

The discussed methods are:

1. Choosing the data location
2. Broadcast encryption
3. Ciphertext Policy-Attribute Based Encryption
4. Data Residency
5. Logging of data location
6. Triangulation

Using sticky policies in combination with an encryption method can be an effective way to enforce data location. Encrypted data can be moved to other countries, as long as the private keys are not transferred. This ensures that the data can only be decrypted in the right places.

It is very important that the keys and the encryption processes are used properly; otherwise both encryption based methods are vulnerable to unauthorized access. In the case of broadcast encryption each entity which can decrypt data can also change the set `S` and re-encrypt the data. In this way it is possible to add other valid decryption keys without consent of the data subject. Fortunately, the data subject is able to observe the changes to the set of accepted entities. However it requires active monitoring by the data subject. In the case of CP-ABE it is possible to generate a new key where the location attribute is set to a false location: an administrator in the US who generates a key with `‘location = Europe’` for a server in the US. This is impossible to detect by the data subject.

Giving customers the possibility to choose the location where data will be stored is quite effective if the contract and SLA give enough assurance that the location settings will be enforced. This method will not give an absolute guarantee that data will not be transferred to other regions. Microsoft Windows Azure claims that it will not transfer data to other regions without explicit permission of the customer, "except where necessary for Microsoft to provide customer support, to troubleshoot the service, or comply with legal requirements ". This may cause violations of local privacy laws.

Using data residency is very effective if all users of an application are within the network of a single organization. Because sensitive data is encrypted or tokenized before it leaves the premises of the customer there is no storage of sensitive data in the cloud. A disadvantage of this method is that it is needed to install a proxy server within the network of the customer. This reduces the accessibility of the cloud service and it also reduces the cost reduction advantage of cloud computing.

Logging movements or storage locations of data can be effective if the cloud infrastructure does not allow movements or storage which circumvents the logging mechanisms.

All methods except triangulation need to be implemented and managed by the CSP. Because the location of data gives no guarantees that certain governments cannot access the data [17], and many CSPs claim the data location is not a very important issue ([19][27]), it is unlikely that CSPs will implement methods which will require large investments, complex software and additional large scale cryptographic key management.

Triangulation of the location of data is the only method which is under control of the customer, so implementation does not depend on a CSP. Question is how reliable this method is. CSPs often have multiple copies of data and do not necessarily deliver this data to users from all data centers, so probably not all copies will be found. Therefore triangulation is a method which can give information about suspected violations by the CSP, but it cannot give guarantees that the data has not been transferred to other regions.

4.5. Conclusions

Completely enforcing that data is only stored in specific regions is only possible by encrypting it on the client side and storing the private keys outside of the cloud. The effect is that CSPs are very limited in what they can do with the data, and many cloud applications will be rendered useless. All methods which give the CSP the possibility to process the data cannot fully enforce the data location.

The described enforcement methods will reduce the risk of data transfer to other regions, as long as they are managed in a proper way. Using such a method increases the confidence customers can have in CSPs when it comes to compliance to data privacy laws.

Table 6 summarizes the characteristics of the methods that were covered in this chapter.

Table 6. The methods to enforce or monitor data location and their characteristics.

	Effectiveness	**CSP effort**	**Customer effort**	**Impl.**	**Control**	**Exists**
Choosing location	Medium. Depends on how the CSP implements the policy.	Low. Most DFSs are already location aware.	Low. Customer only has to choose during set up.	CSP	Customer / CSP	Yes
Broadcast encryption	High. The customer defines which keys can decrypt the data.	High. Key management.	Medium/High. Key management.	CSP	Customer / CSP	No
CP-ABE	High. Key management very important in this case because of the possibility of creating new keys to decrypt data.	High. Key management.	Medium/High. Key management.	CSP	Customer / CSP	Yes
Data residency	High. Sensitive data never leaves the premises.	Medium. Proxy and supporting software have to be developed.	Medium. Needed to install and configure proxy before using cloud service.	CSP/ Customer	Customer	Yes
Logging	Medium. It gives the customer a good tool to audit the storage locations, but it does not enforce anything.	Medium. Storage requirements.	High. The customer has to analyze the logs to find possible violations.	CSP	CSP	Yes
Triangulation	Low. Data is often stored at different locations: hard to trace all copies.	None	High. Different nodes around the globe are needed.	Customer	Customer	Yes

5. Data Deletion

5.1. Introduction

Proper deletion of data is an important aspect of IT operations: data leakage must be prevented and many types of data are surrounded with rules on the maximum retention period. The consequences of data leakages and violations of data retention rules can be significant, as can be seen in the following case:

On 13 June 2012 the Dutch Data Protection Authority (Dutch DPA) announced that it had imposed a penalty on the Dutch Railways for a violation of the national privacy laws (Wet Bescherming Persoonsgegevens, WBP [46]). In The Netherlands a contactless smart card system – called the OV-Chipkaart – is used to handle all the payments for public transport. The computer systems of the different public transport operators store logs of all the travel movements. Regulations state that the maximum retention period for this data is two years. After the maximum retention period all the PII must be deleted. Non-PII may be stored for a longer period for long term analysis, but only if it is impossible to relate the data to a natural person.

After the maximum retention period the Dutch Railways have a procedure to delete the PII which is associated to a travel movement. However, the unique identification number of the chip card is not deleted. Because it is possible to relate an identification number to a natural person using another database the Dutch DPA ruled that the deletion procedure is not in compliance with the WBP. In addition an investigation concluded that the backups of the travel logs were stored for a longer period than allowed. The Dutch Railways received a penalty of €125,000 and the PII is deleted.

The risk of insecure or ineffective data deletion is classified as one of the top risks by the ENISA report [2].

On-premise IT models offer a company full control over all the storage media so it is possible to overwrite data several times or to physically destroy a storage medium when data has to be deleted. With cloud computing using those methods is often not possible because a company does not have full control over storage media in the cloud and storage media are shared between customers. Many cloud storage systems use replication and versioning systems and it is not clear where the data is stored. So even if a CSP implements algorithms to properly delete data from storage media it is very hard to guarantee that all the copies of the data will be deleted.

In section 5.2 several deletion methods will be described. Because a deletion method by itself is not enough to get control over the whole deletion process, a new method which tracks copies of data within a cloud infrastructure and uses this information to delete all copies will be introduced in section 5.3.

5.2. Deletion Methods

5.2.1. Disk wiping and physical destruction

When data is deleted from a hard disk the storage space is only marked as available so it can be overwritten with a new file. This means that the data is still available if it is known where it was stored on the disk. If the data needs to be deleted in a proper way it is possible to overwrite the data several times with random bits [28].

The storage and database systems of CSPs abstract away from the low level functionality of the storage methods. Data is being replicated and previous versions are not deleted to provide rollback functionality. It is very hard to know which parts of which hard disks have to be erased.

Effectively using physical destruction of a hard disk to delete specific data is even harder, because data of different customers is stored on a single disk. CSPs often physically destroy hard disks if they are no longer used in the cloud infrastructure.

5.2.2. Crypto-shredding

Encrypted data can be deleted by destroying all the decryption keys. This method is called crypto-shredding [29]. The method works as long as the used encryption method is so strong that it is infeasible to break it. The main advantage of this method is that it does not matter how many copies of the data the CSP stores; if all copies are encrypted and the keys are destroyed all copies are deleted. Deletion can also be very fast because deleting the keys often involves much less data than deleting complete files.

The main disadvantage is that in most cases the keys are known to the CSP, so the cloud service can access the data. The CSP has to store the keys on their infrastructure, so it becomes very hard to track how many copies there are of keys and where they are stored. Proving that all copies of a key are deleted in an irreversible way is thus very hard to do.

5.3. A data provenance based method for data deletion

Data storage and processing often results in numerous copies of the same data. File systems automatically replicate data, back-ups are made, old versions of documents are stored in an archive, new data is derived from other data, data is used in ETL processes, etc.

Properly deleting all copies of a data artifact can be very hard if it is not known exactly which copies there are. Fortunately there are methods that track from which artifacts an artifact was derived by which process. These so called data provenance methods answer the questions “Where does this data come from?” and “How was it created?” [30][31].

Data provenance methods implement the tracking of the ancestry of data as a *directed acyclic graph* (DAG) with edges that indicate from which artifacts an artifact was derived. If the direction of the edges is reversed the resulting graph holds information on which artifacts were derived from a specific artifact. This can be used to effectively find all the copies of some data and thus it can be used to find all artifacts that have to be deleted to effectively remove data.

This idea will be expanded in the rest of this section. We will start with introducing a formalized model of data provenance in 5.3.1. Next the complete concept is discussed in 5.3.2, and the corresponding algorithms are given in 5.3.3.

5.3.1. The Open Provenance Model

The *Open Provenance Model* (OPM) [30] is a model for the collection, storage and exchange of provenance information. OPM describes data provenance in the form of a graph with different types of nodes and different types of edges.

There are three types of nodes. *Artifacts* are pieces of data which are tracked by the provenance system. A *process* is an action or a series of actions on one or more artifacts; a process will result in one or more new artifacts. An *agent* is entity which acts as a catalyst for processes. In Figure 11 a graphical representation of the different nodes is shown.

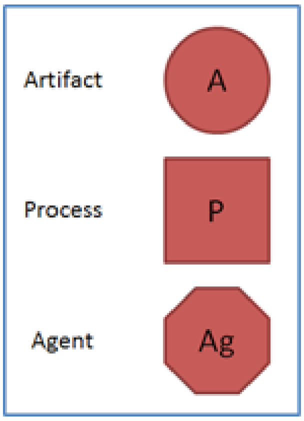


Figure 11. OPM Nodes

To connect the nodes five types of edges are used.

- **used:** Indicates that process *P* used artifact *A*,
- **wasGeneratedBy:** Indicates that artifact *A* was generated by process *P*,
- **wasControlledBy:** Indicates that process *P* was controlled by agent *Ag*,
- **wasTriggeredBy:** Indicates that process *P2* was triggered by process *P1*,
- **wasDerivedFrom:** Indicates that artifact A2 was derived from artifact *A1*.

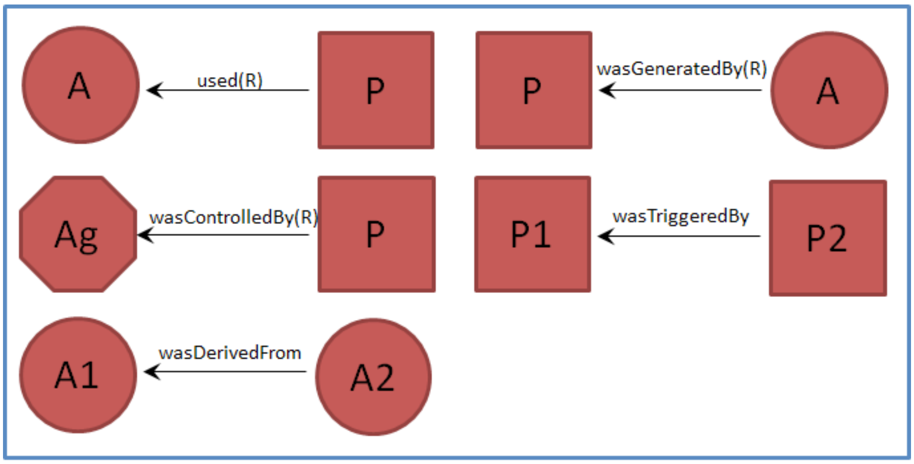


Figure 12. OPM Edges

In Figure 13 an example of an OPM graph is shown. It describes the process of baking a cake (*bake*). This process used *100g butter, two eggs, 100g flour and 100g sugar* (the dotted edges represent used-edges). The process was controlled by *John*. The result was a *cake*. The solid edges specify from which ingredients the cake was derived (wasDerivedFrom-edges).

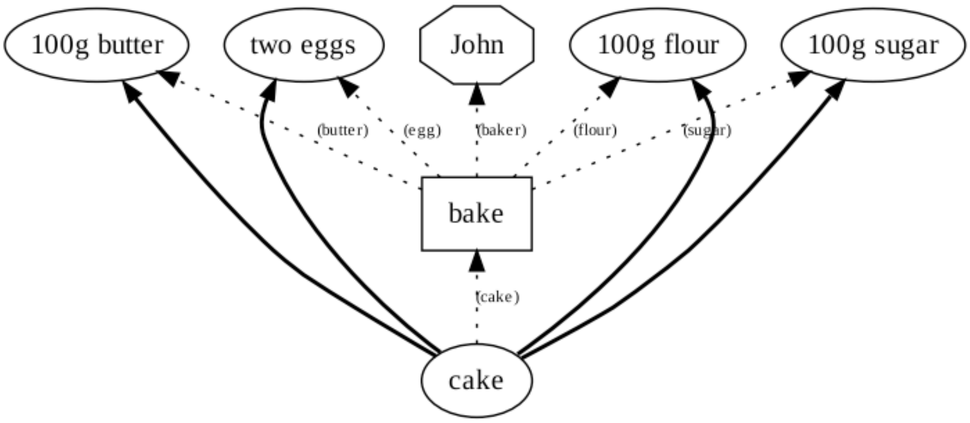


Figure 13. Example of an OPM graph (from [30])

5.3.2. General idea

If a CSP implements a data provenance system one can ask the question from which artifact a certain artifact was derived. It can be answered by following the wasDerivedFrom edges from that artifact. Because the start and end points of all the wasDerivedFrom edges are known it is also possible to reverse these edges, so it is possible to find out which artifacts were derived from a specific artifact. By repeating this step for all derived artifacts until no new artifacts are found a graph containing all artifacts which were derived from a specific artifact can be constructed.

Such a graph gives an overview of where data is stored throughout the cloud infrastructure. Besides that, such a graph can be used to determine which artifacts have to be deleted if some data has to be deleted. The deletion algorithm will delete or process the descendants in a recursive way, after which it can process the *'root'* artifact. While this approach guarantees that all copies that are known to the provenance system will be deleted, it is often an approach which is too drastic. Many artifacts are constructed from more than one artifact, so deleting a complete artifact might destroy more data than desired.

To address this problem, wasDerivedFrom edges can be annotated with an instruction on which action the algorithm has to take on the artifact at the other end of the edge. The actions that the algorithm can perform on an artifact are:

- **delete:** Completely delete the artifact,
- **partial delete:** Scan through the artifact and only delete data which needs to be deleted,
- **anonymized:** Anonymize (parts of) the data, so it is still usable but there is no sensitive information left,
- **none:** Keep the artifact intact and do not perform actions on artifacts which are derived from the artifact.

The algorithm will report which actions it has taken on which artifacts, so a customer can inspect if the data was deleted in a proper way. It is also possible to check if all relevant artifacts were deleted by trying to retrieve them from the cloud application. If all requests fail or do not contain the data that needs to be deleted, the deletion operation was successful.

In the next subsection a more formal description of the annotations and the algorithm will be given.

5.3.3. The algorithm

Before the issuer of the deletion operation starts with deleting the artifact he retrieves the provenance graph of all artifacts that were derived from the artifact that has to be deleted. This information is needed to be able to verify if the deletion algorithm performed the right operations.

Retrieving the provenance graph for artifact `A` is done by calling `GetTree(A)` (Figure 14), which is a method that creates a new `Graph` object which will store the result and calls another method `GetTree(A, graph)`.

`GetTree(A, G)` (Figure 15) adds the current root node `A` to the list of nodes of the graph and retrieves all provenance edges between `A` and other artifacts which were derived from `A`. These edges are added to the list of edges of the graph. Next the method does a recursive call to start the processing of the artifacts which were derived from `A`. When all calls to `GetTree(A, G)` return `GetTree(A)` will return the complete provenance graph for artifact `A`.

```
GetTree(A):

    Graph graph := new Graph()
    GetTree(A, graph)

    return G
```

Figure 14. `GetTree(A)` retrieves the provenance graph with root artifact `A`.

```
GetTree(A, G):

    G.nodes.add(A)
    edges := GetProvenanceEdges(A)
    G.edges.append(edges)
    for each edge in edges:
        GetTree(edge.dest, G)

    return G
```

Figure 15. `GetTree(A, G)` adds nodes and edges to the provenance graph for the sub graph which is rooted at `A`.

Deleting an artifact `A` starts with calling `DeleteArtifact(A)` (Figure 16). This method initializes a `Log` object which keeps track of all the actions that are performed by the algorithm so it can be checked if the algorithm performed the right actions. Next the set of all edges between artifact `A` and artifacts that were derived from `A` are retrieved. For each edge `ProcessEdge` is called, which recursively handles the graph with starting point `edge.dest`. After the algorithm is finished with processing the derived artifacts `DeleteArtifact` will `Delete A` and add the result of that operation to the log.

```
DeleteArtifact(A):

    Log log := new Log()
    edges := GetProvenanceEdges(A)
    for each edge in edges:
        ProcessEdge(edge, log)

    log.add(Delete(A))
    return log
```

Figure 16. `DeleteArtifact(A)`, the top level method of the algorithm which starts the graph traversal and deletes artifact `A`.

`ProcessEdge(E, log)` (Figure 17) takes an edge `E` and a log object `log` as parameters. It starts with checking if the action annotation does not indicate that the algorithm should not propagate past the destination artifact of the edge. If the algorithm should propagate `ProcessEdge` retrieves all the provenance edges from `E.dest` to derived artifacts and calls `ProcessEdge` for all the retrieved edges.

After the derived artifacts are processed `ProcessEdge` determines which action it has to take on the current artifact `E.dest`. This can be `‘delete’`, `‘partial_delete’` or `‘anonymize’`.

The `Delete`, `PartialDelete` and `Anonymize` methods are methods that need to be implemented for the specific application where this algorithm is used.

When the algorithm is finished with traversing the graph `DeleteArtifact` returns the log object which contains a list of all actions that were performed by the algorithm. This list can be used by the issuer of the deletion operation to verify whether the data was deleted properly. This can be done by comparing the list with the provenance tree that was retrieved before the deletion operation.

```
ProcessEdge(E, log):

    if E.annotations.action != ‘none’:
        edges := GetProvenanceEdges(E.dest)
        for each edge in edges:
            ProcessEdge(edge, log)

    switch E.annotations.action:
    case ‘delete’:
        log.add(Delete(E.dest))
    case ‘partial_delete’:
        log.add(PartialDelete(E.dest, E.src))
    case ‘anonymize’:
        log.add(Anonymize(E.dest, E.src))
```

Figure 17. `ProcessEdge(E, log)` handles the artifact `E.dest` and recursively processes the part of the graph which originates from `E.dest`.

5.3.4. Adapters

The processing methods for data artifacts are very general. `Delete`, `PartialDelete` and `Anonymize` do not specify how they are implemented. Because the underlying storage methods and artifact types determine which actions have to be taken the processing methods will be polymorphic. For example, the `Delete` method needs to be implemented for a file, a record and other types. Methods with multiple parameters can be defined for any combination of types.

```
Delete(File artifact)
Delete(Record artifact)
Anonymize(Record dest, Record src)
Anonymize(Record dest, File src)
```

Figure 18. Examples of polymorphic artifact processing methods.

The specific implementations of the processing methods need to be done by the developers of an application that will use the provenance based deletion method.

5.3.5. An example

In the left part of Figure 19 a provenance graph with six artifacts and their corresponding provenance edges is shown. *A3* was derived from *A1* and *A2, A4* and *A5* from *A3* and *A6* from *A5*. If *A1* or *A2* will be deleted *A3* also has to be deleted. If *A3* is deleted *A4* will be anonymized and no action will be taken against *A5* and *A6*. If *A5* is deleted *A6* is also deleted.

Now the algorithm is started by calling `DeleteArtifact(A1)`. During execution it performs the following steps (where *<A2,A1>* represents the edge that indicates that *A2* was derived from *A1*):

1. `DeleteArtifact(A1)` calls `ProcessEdge(<A3,A1>, log);`
2. `ProcessEdge(<A3,A1>, log)` calls `ProcessEdge(<A4,A3>, log);`
3. As there are no provenance edges that point to *A4* `ProcessEdge(<A4,A3>, log)` anonymizes *A4*, logs the result and returns;
4. `ProcessEdge(<A3,A1>, log)` calls `ProcessEdge(<A5,A3>, log);`
5. Because *<A5,A3>* has `none` as its action annotation `ProcessEdge(<A5,A3>, log)` does nothing and returns;
6. `ProcessEdge(<A3,A1>, log)` deletes *A3*, logs the result and returns;
7. `DeleteArtifact(A1)` deletes *A1*, logs the result and returns the log to the caller.

The resulting provenance graph is shown in the right part of Figure 19.

Note that the resulting graph is not connected. This is undesirable in cases where the provenance information is also used for other purposes. This problem can be solved by preserving the edges or by creating new edges between the remaining artifacts (*<A4,A2>* and *<A5,A2>* in this case).

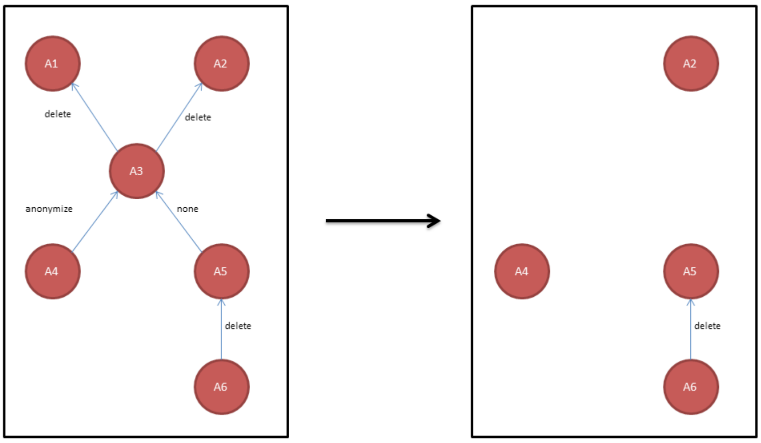


Figure 19. A provenance with six artifacts and corresponding provenance edges. The result of deleting artifact A1 is shown in the right provenance graph.

5.3.6. Simulation

To test the algorithm it is implemented in a simple test script written in Python[1]. The artifacts and wasDerivedFrom edges are represented by a networkx[2] graph structure which can be manipulated using an API. With this graph structure and API the methods of the algorithm are implemented and tested. The code of the simulation program can be found in “Appendix A. Deletion algorithm”.

As a demonstration of the simulator the test case from the previous subsection is simulated. First, `GetTree(“A1”)` is called to verify the structure of the provenance graph. The result can be seen in Figure 20.

```
Node(A1)
Edge(A3, A1) with action delete
Node(A3)
Edge(A5, A3) with action none
Node(A5)
Edge(A6, A5) with action delete
Node(A6)
Edge(A4, A3) with action anonymized
Node(A4)
```

Figure 20. The result of running `GetTree("A1")`. This structure corresponds to the graph in Figure 19 (left).

Next, `DeleteArtifact(“A1”)` is executed and the returned log (Figure 21) is inspected. It can be seen that the algorithm behaves as expected.

```
Anonymized artifact A4
Deleted artifact A3
Deleted artifact A1
```

Figure 21. The log items that are returned by running `DeleteArtifact("A1")`.

---

[1] Python: http://www.python.org/

[2] NetworkX: http://networkx.lanl.gov/

Because the provenance data is not connected after `DeleteArtifact(“A1”)` the raw lists of nodes and edges are printed. From these lists one can infer that the resulting graph has the same structure as the right part of Figure 19.

```
[‘A2’, ‘A5’, ‘A4’, ‘A6’] (nodes)
[(‘A6’, ‘A5’)] (edges)
```

Figure 22. The resulting graph after the execution of `DeleteArtifact("A1")`.

5.4. Discussion

We have seen that proper data deletion is important. Many laws and policies require companies to delete specific data, but often there are numerous copies of data which are hard to track. Deleting all copies is thus also hard to do as we have seen in the case of the Dutch Railways. Especially in the cloud a company has less control over the storage media where its data is stored. This calls for effective methods to delete data.

If it is clear what has to be deleted the deletion operations should delete the data in an irreversible way. This can be done using disk wiping, physical destruction or crypto-shredding. The first two methods are problematic to use in the cloud to delete data of a specific customer because data of different customers is stored on the same storage medium. The former method overcomes this problem; however it depends on the effectiveness of the key destruction methods.

These three methods can be very effective; however it is next to impossible for a CSP to prove to a customer that data has been deleted.

In many cases it is not clear how many copies there are of a specific data artifact; there are many ways in which copies of data can be made. To address this problem this chapter suggested the use of data provenance to determine which artifacts have to be deleted or altered. If the provenance metadata is collected in a consistent way, the method can track all copies of a data artifact which are within the control boundaries of the CSP. The proposed algorithm uses the provenance graph to recursively delete a data artifact and all other artifacts that were derived from that data artifact.

Deleting data is not as easy as deleting all data which was derived from that data. Artifacts can be derived from different other artifacts, some policies require that data is stored for a longer duration and anonymized data can be used to perform statistical analysis. To support these use cases the provenance data can be annotated with action parameters which specify which action the deletion algorithm has to take.

With this algorithm there is still no absolute proof that the data is gone, but the methods that allow a user to see the governance graph and that give a list of all actions the algorithm has performed make the deletion process more transparent and thus easier to perform audits on.

The adoption of data provenance and data provenance based deletion methods can be quite hard because it takes much storage space and it is not trivial to implement. However, with the increasing focus on accountability and matters such as privacy laws that dictate maximum retention periods, data provenance may become a necessity.

5.5. Conclusions

To perform proper data deletion from storage media methods like disk wiping and crypto-shredding are needed. Because there are many cases where it not completely clear what has to be deleted a data provenance based method is described which uses the provenance information to determine which data artifacts have to be deleted or altered. The method also allows a user to inspect the actions taken by the deletion algorithm, which makes data deletion more transparent.

Although the implementation of data provenance and this method require a considerable amount of effort and storage, matters such as accountability and privacy laws may increase the adoption of data provenance and data provenance based deletion methods.

In Table 7 the characteristics of the discussed data deletion methods are summarized.

Table 7. The characteristics of the discussed data deletion methods.

	Effectiveness	**CSP effort**	**Customer effort**	**Impl.**	**Control**	**Exists**
Disk wiping	Low. Data of multiple customers on single disk. Only usable for end of life destruction of disk.	Per customer: High. End of life: Medium	Low	CSP	CSP	Yes
Crypto-shredding	High. With proper key management all copies of a data artifact can be deleted instantly.	Crypto already in use: Low	Low	Customer / CSP	Customer / CSP	Yes
Provenance based deletion	High. Clear which copies of data exist and which copies need to be deleted. Highest effectiveness in combination with crypto-shredding.	High. Necessary to track all data operations.	Medium	CSP	Customer / CSP	No

6. Data Leakage

6.1. Introduction

Leakage of data to external parties is a serious risk in traditional computing as well as in cloud computing [2][10]. The Open Security Foundation keeps track of reported data leakage incidents on a special website (http://datalossdb.org/). Last year 1041 incidents were reported. Such an incident can have a very big impact on an organization in the form of financial and reputational loss and legal problems.

Mitigating the risk of data leakage is difficult because there are different types of adversaries. Data can be leaked by internal or external persons, and leakage can be intentional or by accident (see Table 8). Figure 23 shows a breakdown of data leakage incidents by type of adversary or vector.

Table 8. Examples of data leakage scenarios with internal and external adversaries which leak data intentionally or by accident.

	Intentional	**By accident**
Internal	E-mailing a secret document to a competitor	Losing an unencrypted USB-stick with sensitive data
Internal	Bypassing security measures in a computer system to gain access to internal data	Stumbling upon confidential documents which were placed on a public facing website by accident

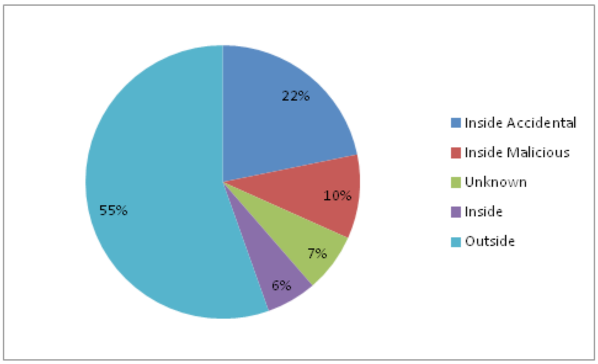


Figure 23. Incidents by vector (source: www.datalossdb.org)

Especially difficult is the protection of data against leakage by internal users with permission to access that data. It is not suspicious if an authorized user accesses data, so only when that user tries to leak data detection is possible. With on-premise IT infrastructure and managed workstations and laptops it is possible to use technologies like traffic inspection at the network boundary between the company's internal network and the internet. Another widespread practice is to install endpoint software which monitors user activity.

With cloud computing these methods are less effective because data needs to be transferred between the CSP and the user before it can be used. This means that it is not possible to monitor traffic at a network boundary. With emerging Bring Your Own Device (BYOD) possibilities companies also have less control over the devices that are used to access the cloud application, so it is not always possible to install endpoint protection software on all devices that have access to the application.

External attempts to gain access to data in the cloud can be performed in many different ways. Each security flaw in the cloud software stack can potentially lead to data leakage [32].

In the next sections several cloud specific technologies to prevent and detect data leakage will be discussed.

6.2. Prevention

Preventing data leakage by insiders starts with ensuring that users can only access data that they need to perform their work. This does not directly protect against data leakage, but it reduces the amount of data a single person is able to leak. Because it is not possible to prevent data from leaving the cloud infrastructure, methods that prevent data leakage should reduce the amount of useful data that can be leaked. This can be done by masking the data (6.2.1), enforcing access controls using cryptography and trusted computing (6.2.2) or polluting a data set with fake data (6.2.3).

6.2.1. Data Masking

Reducing the amount of sensitive data that can be leaked is an effective approach to reduce the impact of data leakage incidents. But just hiding the information is not always possible. Software development often involves separate development, testing, acceptance and production environments. Users of the production environment have – for example – access to customer records which contain addresses and credit card numbers. But users of the testing environment (testers) also need realistic data to properly test the application. Using production data for testing purposes increases the risk of data leakage because the data is available to more users. Because the most important requirement for activities like testing is that the data is realistic it is possible to modify the data in a way which makes the data less sensitive. This can be done in several ways:

- **Substitution:** Sensitive data can be replaced with (randomly) generated values. For example, credit card numbers can be replaced with randomly generated valid credit card numbers.
- **Shuffling:** Randomly replacing sensitive values with values from other records. For example, a telephone number can be replaced with the telephone number from a random person in the database.
- **Number or date variance:** Adding (small) random numbers to numerical data.
- **Nulling out or deletion:** Completely removing sensitive data from records.
- **Masking out:** Replacing (parts of) a value with ‘*’, ‘#’ or another character. For example, credit card numbers need to be (partially) masked out if a user does not have a legitimate business need to see the full credit card number [33].

These methods can be applied on the whole dataset before it will be used (static data masking) or on specific records at the moment they are requested (dynamic data masking). It is also important to select a data masking method which makes it infeasible to reverse the masking operation. Shuffling and number or data variance can be reversed if the masking operation is not ‘random’ enough.

6.2.2. Trusted Computing

In [34] and [35] Alawneh and Abbadi introduce a method for protecting sensitive documents against leakage incidents. The method uses encryption and the concepts of Trusted Computing[1] to regulate which users and machines within and outside the trust boundary of an organization have access to files. Because the method can transcend trust boundaries it is potentially an effective technological measure to overcome an inherent problem of cloud computing: data needs to be transferred to clients which are not inside the control boundary.

Sensitive documents can be shared within a dynamic domain. A dynamic domain is a collection of devices and it has an identifier i_D and a symmetric key k_D which is used to perform cryptographic operations on documents which need to be shared within the dynamic domain. The dynamic domains and keys are managed by a master controller.

Each device needs to have a *Trusted Platform Module* (TPM), which is a chip that can handle cryptographic operations and securely store cryptographic keys. When a device becomes a member of a dynamic domain its TPM needs to receive and store the i_D and k_D. Because a TPM is a specialized hardware component with tight security measures it can be used to protect the key against malicious use.

A device retrieves encrypted documents from the master controller, and can only decrypt the document if the device is member of the right dynamic domain. Modified or new documents can be encrypted with the key of the desired dynamic domain before sending the document back to the master controller.

6.2.3. Fog Computing

In [36] Stolfo et al. propose a new approach for preventing data leakage from cloud applications. Data access patterns are monitored for abnormal user behavior to detect user masquerading. Because this method can lead to more false positives than desired, it is extended with decoy information. These documents are genuine looking documents, but they do not contain real data. A normal user who is performing his normal tasks will never access these files, but a masquerading user – who will most likely have less knowledge about the structure – has a significant chance of accidentally touching these decoy files. With the combination of these two approaches it is possible to detect masquerade attacks with a false positive rate of 1.12%.

Monitoring user behavior is done by analysis of the file search operations by the users. In most cases a normal user will issue targeted and limited search queries, while a masquerading user will perform much wider search operations because he has less knowledge about the file system structure.

The decoy method tracks if the decoy files (or honeypots) are accessed by a user. If a user accesses one or more decoy files and the search behavior monitoring tool also classifies the user's behavior as suspicious the software will raise an alert and it will poison the data which is sent to the user with large amounts of legitimate looking fake data. This will diminish the value of the leaked data.

---

[1] Trusted Computing Group: http://www.trustedcomputinggroup.org/

6.3. Monitoring

While fog computing is a combination of data leakage monitoring and prevention, the following two approaches are only able to detect data leakage incidents. In Log file analysis (6.3.1) the very general approach of analyzing log files to find strange behavior will be discussed. In Psychological triggers (6.3.2) a more specific log analysis approach which promises to be more accurate and less storage demanding is discussed.

6.3.1. Log file analysis

A data leakage monitoring method which requires a relatively small amount of software at the side of the CSP is logging all file or record read, create, update and delete operations. These logs can be analyzed by external tools. This approach is especially suitable for detecting incidents in which large amounts of data are transferred, because the analysis techniques focus on finding deviant behavior of users. If a user only leaks a single document during office hours when it is normal for him to access such kinds of documents, it is very hard to detect this event. But if a user opens a document outside office hours or if a user suddenly opens a large amount of files, it can be a clue that unauthorized actions are performed on the data.

6.3.2. Psychological triggers

In [37] Sasaki proposes a new method for detecting insider threats using psychological triggers. This method should have significant advantages over existing analysis methods for detecting suspicious behavior. These advantages will be achieved by:

- Reducing the number of false negatives,
- Reducing the number of false positives,
- Covering more use cases,
- Reducing the storage requirements.

Sasaki argues that it is possible to trigger alternate behavior by malicious insiders and use detection methods to find those users. A trigger can be a companywide announcement that an investigation will start. It is likely that users which have something to hide from the investigators will stop their malicious activities and try to destroy evidence by deletion or altering files, e-mails and other data.

Before the users are triggered an agent which logs all actions by all users will be started. After the trigger is ‘fired’ the logger will continue. When the logger is terminated it is possible to analyze all actions and see which users have a significant difference in behavior before and after the trigger.

This method should be better than the existing methods because it addresses the four requirements in the following way:

- **False negatives:** A data leakage might be very hard to detect because a malicious user can leak data which he already had to use to perform his job and the leaking can be performed on untrusted machines. Deletion of evidence is easier to detect because it is less likely that these actions will blend into the normal behavior of a user.

- **False positives:** The number of false positives is the false positive rate multiplied by the number of events. So, if the number of events is reduced the false positive rate is also reduced. By only monitoring user behavior just before and during the 'investigation' there will be much less events in comparison to continuous monitoring methods.
- **Use cases:** This method detects psychological reactions instead of specific behavioral patterns that are associated with leaking data, stealing money, espionage, etc.
- **Storage requirements:** The storage requirements are reduced because the logging tool will only be activated during a limited amount of time.

In cloud computing applications this method can be employed by using a subset of the audit trails.

6.4. Discussion

The methods that were discussed in the previous sections all try to prevent or detect data leakage incidents. A single method will not be able to completely remove the risk, but with knowledge of the most important advantages and disadvantages of the discussed methods it is possible to assess their effectiveness in different scenarios.

6.4.1. Data Masking

We have seen that data masking techniques are able to reduce the amount of sensitive data that can be leaked from an application. Especially in development and testing environments the most important requirement on data is that it is realistic, so masking sensitive values with other (random) realistic values significantly reduces the risk of data leakage.

6.4.2. Trusted Computing

The work of Alawneh and Abbadi [34][35] promises good protection against data leakage by using dynamic domains, cryptography and TPM technology. Using this method it should be possible to reduce the risk of (inadvertent) data leakage. However, because there is no complete control over the device and the other software which is running, it is not possible to prevent actions such as copying parts of a document to an unprotected document. So, this method makes (especially inadvertent) data leakage harder, but certainly not impossible.

Another problem with this method can arise when devices are lost or stolen and do not have network connectivity, because the master controller needs to communicate with devices to revoke keys. Thus, if encrypted documents are stored on the device it is still possible to decrypt those documents until a network connection is established. When the method is used in cloud computing this problem will not have a big impact because in-browser cloud applications reload data on each request.

6.4.3. Fog Computing

Fog computing is a method which tries to prevent data leakage by masquerading persons. The assumption that a masquerader knows less about the structure of the file system implies that he is not an internal user who uses the account of a colleague.

It might be possible to extend fog computing by using a machine learning algorithm which learns what the normal behavior of each user is. A classifier can then detect if the current usage patterns significantly differ from the normal usage patterns, and thus detect if an internal user is masquerading. Question is if a disinformation attack is effective in this case because an internal user might easily notice whether data is fake.

6.4.4. Psychological triggers

The reduced timespan in which investigations using this method are done has the advantages that less storage is needed and that there will be less false positives. A disadvantage is that an organization must suspect that something happened, because an investigation has to be announced. If there are no suspicions there will be no investigation.

6.5. Conclusions

We have seen that no single method can completely prevent data leakage by internal persons. Especially in cloud computing environments many classical protective technologies cannot be used because not all devices are under control of an organization. The most effective way to reduce the risk of data leakage is to reduce the amount of information a single user can leak; most of the discussed methods focus on accomplishing this.

Table 9. The characteristics of the discussed data leakage prevention and detection methods.

	Effectiveness	**CSP Effort**	**Customer effort**	**Impl.**	**Control**	**Exists**
Data masking	Medium. Some users still need access.	Medium	Low	CSP	CSP	Yes
Trusted computing	Medium. Restricts access to specific users, but copy-paste cannot be prevented.	Medium	High. All devices need TPMs and management is difficult.	Customer / CSP	Customer / CSP	Yes
Fog computing	Medium. Completely depends on detection algorithm and amount of disinformation.	High. Analysis of all data operations.	Low. Customer receives alerts.	CSP	Customer / CSP	Yes
Log file analysis	Low. Large amounts of data and abuse can be too subtle to notice.	High. Need to find patterns in large amounts of data.	Low. Customer receives alerts.	CSP or Customer	Customer / CSP	Yes
Psychological triggers	Medium. Less data to analyze and triggering of suspicious behavior. Suspicion needed to start an investigation.	Medium. CSP only needs to supply log files.	High. Customer or other party needs to investigate.	Customer (and CSP)	Customer	Yes

7. Data segregation

7.1. Introduction

The segregation of data from different customers is a very important requirement of cloud services. Incidents where customers can access data from other customers can diminish the trust which customers have in the service. Because it is such an important aspect CSPs put a lot of effort in getting data segregation right. The amount of documented cases is low (the footnotes contain some examples[1] [2]).

The segregation of data of different customers needs to be enforced on all levels. If data is only properly segregated while it is stored on hard disk, attackers may access data from other customers during data transfer or processing. To cover all aspects of data segregation the three states of data model will be used [38]. This model states that data can be in one of three states:

- **Data at Rest (DaR):** Data that is stored in a database or file system (storage).
- **Data in Motion (DiM):** Data that is being transferred from a database or file system to a place where it will be processed (networking).
- **Data in Use (DiU):** Data that is currently being processed by an application and processor (computing).

In the next three sections the different methods which can be used to segregate data in the different states will be discussed.

7.2. Data at Rest

7.2.1. Databases

Multi-tenant databases can be separated using three different approaches. The approach with the highest isolation is using a separate database for each customer. Using separate schemas within a single database introduces more flexibility, but it reduces isolation. With the shared schema approach it is easier to support large numbers of customers. However, with shared schema the developers of the application are completely responsible for the correct implementation of the segregation controls [39].

7.2.1.1. Separate databases

Giving each customer separate databases and database authentication credentials offers the most isolated data segregation method because data is logically separated by the database server software. Besides the proven security controls of database systems using separate databases also brings two other advantages. First, it is less difficult to alter the data model on a per customer basis.

Second, it is easy to restore backups for a specific customer than with other methods where data of different customers is stored in the same database.

---

[1] http://www.scmagazine.com/Google-Docs-flaw-could-allow-others-to-see-personal-files/article/116703/?DCMP=EMC-SCUS_Newswire

[2] http://www.csoonline.com/article/487089/analyst-spots-three-flaws-in-google-docs

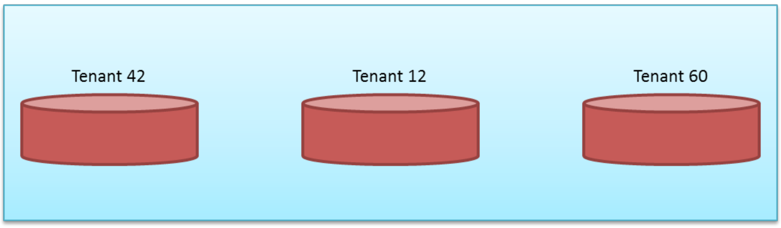


Figure 24. Data segregation based on separate databases.

The downside of this approach is that it is a relatively expensive approach. Often there is a maximum number of databases which can be hosted on a single database server, and thus the cost of hardware and server administration increases.

7.2.1.2. Separate schemas

A schema is a logical collection of tables which can have specific (security) settings. Specifically, it is possible to restrict user access to specific schemas instead of specific databases.

By using separate schemas for data segregation it is possible to service more customers with a single server, and thus it is a cheaper approach than using separate databases. The data model flexibility advantage of separate databases still holds for separate schemas. A disadvantage is that restoring backups is harder because only a part of a database needs to be restored.

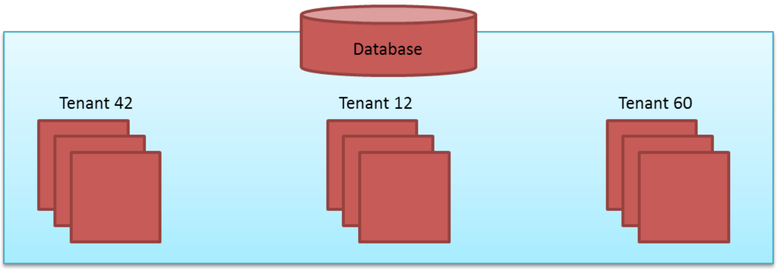


Figure 25. Data segregation based on separate schemas within the same database.

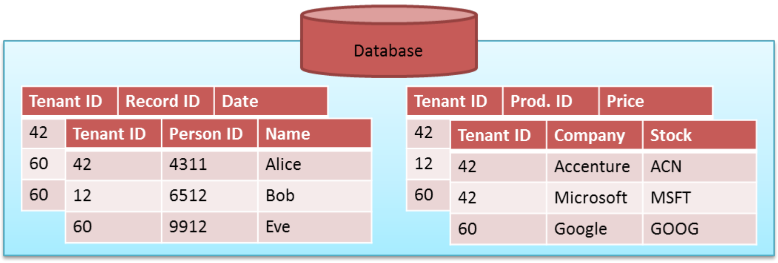
7.2.1.3. Shared schema

Storing the data from different customers in the same database and the same schema offers the biggest scalability advantages. However, because a type of record is stored in the same table for different customers and database servers have no native support for data isolation at this level, it is the responsibility of the CSP to implement effective isolation controls. The most used method to implement these controls is to associate a tenant ID with reach record. When a query is executed an extra WHERE clause will ensure that only records with the tenant ID of the current user are returned.

7.2.2. File storage

7.2.2.1. Authentication and authorization

To ensure that only the right users can access files in the cloud, the cloud service needs to implement strong authentication and authorization mechanisms. Authentication mechanisms are needed to verify the identity of the current user. In the context of data segregation authorization mechanisms are needed to ensure that the user can only access files that are owned by that user or his employer.



Tenant ID	Record ID	Date
42		
60		
60		

Tenant ID	Person ID	Name
42	4311	Alice
12	6512	Bob
60	9912	Eve

Tenant ID	Prod. ID	Price
42		
12		
60		

Tenant ID	Company	Stock
42	Accenture	ACN
42	Microsoft	MSFT
60	Google	GOOG

Figure 26. Data segregation based on a shared schema.

Authorization controls can be implemented in the underlying file system of a storage service or in the service that handles the access to the stored files. If file system level controls are used the CSP can reduce its development efforts and files are also isolated in the file system. If the controls are in the service software the CSP has to develop them, which results in more work and possibly increased security risks. The advantage of this approach is that a cloud service user account does not have to correspond to a user account in an underlying file system, so a file system can be simple and more scalable.

7.2.2.2. Encryption

Encrypting the data of a customer with a customer specific key can also reduce the data segregation risk. If the cloud service needs access to the data the key needs to be stored in the same infrastructure, so in this case data segregation depends on the secrecy of cryptographic keys. In cases where data is only stored in the cloud and the cloud services do not need access to the data the key can be kept out of the cloud. In this case the segregation of user data is very strong.

The access of cloud services which need to process data to decryption keys poses increased security risks. This problem can be reduced if it is possible to use and manipulate data without decrypting it. Homomorphic encryption (HE) is a relatively new encryption paradigm which can be used to achieve that. The idea is to take an operation on a specific type of data and to find an equivalent (homomorphic) operation which acts on encrypted data.

Unfortunately there are only equivalent operations for a limited number of normal operations. HE is also computationally expensive. Therefore, HE is not yet mature and cannot be used for practical purposes.

```
Operation(Data) = Data'
CryptoOperation(E(Data)) = E(Data')
```

Figure 27. If `Operation()` and `CryptoOperation()` are homomorphic `CryptoOperation()` can be applied to encrypted data to perform the same computation as `Operation()` and produce the same result in an encrypted way.

7.3. Data in Motion

Especially in the case of IaaS clouds data segregation at the network level is important. With IaaS the tenants have (limited) access to the underlying network from their VMs. If the network is not properly secured it might be possible to intercept other tenants' communications.

7.3.1. Virtual Local Area Networks

Placing all (virtual) devices from a specific tenant in a separate *Virtual Local Area Network* (VLAN) can effectively protect network communication. In a VLAN each packet is tagged with a VLAN identifier. Existing physical and virtual network devices are aware of which connected devices belong to which VLANs, and segregate packets from different VLANs.

Unfortunately there is a limit of 4k VLANs in a single network environment, which severely affects the scalability of using VLANs to segregate different tenants' moving data. To make it scalable Hao et al. [40] propose an architecture which separates a cloud infrastructure into different domains, which each can contain the maximum number of VLANs. Each domain that contains VMs of customers is called an *edge domain*, and there is one *core domain* which facilitates and coordinates the communication between edge domains and between edge domains and the internet. The global structure of such a network can be seen in Figure 28.

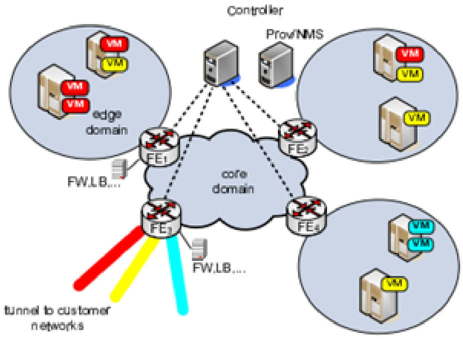


Figure 28. The architecture from [40].

The boundary of an edge domain with the core domain is formed by one or more *Forwarding Elements* (FEs). They determine to which other FE a packet needs to travel. After this is determined an FE tunnels a packet to another FE. In this way a virtual network of a single customer can span multiple VLANs in different edge domains.

Another possibility of this approach is that it becomes possible to treat the internal network of a customer as another edge domain. The customer can connect using VPN technology, and the VPN gateway in the cloud acts as a FE. Using this approach it is possible to create a single virtual network from a cloud based virtual network and the physical network of the customer.

7.3.2. Encrypted communication

The communication between two (virtual) machines within the cloud infrastructure can also be secured by using for example TLS or SSH tunneling in the application layer or IPSec in the internet layer. This works well in the case of IaaS where users have access to the network.

With SaaS or PaaS customers do not have dedicated VMs, and network connections are not always specific to a customer. Therefore it is not feasible to use the discussed low level methods. To protect data in motion in these scenarios stored encrypted data should be decrypted after it is transferred.

7.4. Data in Use

While a virtualization platform separates the processes and data of different customers hardware resources are shared. On a single machine a single processor core handles the work of several tenants. Under normal circumstances different processes will be separated in an effective way, but there are attack vectors like cache missing [41] which can be used to leak data from another process.

Because compromising data segregation at the data in use level happens by running custom attack code at a low level there are two possible approaches which can be used to reduce this risk.

The first approach is improving the processor designs and settings. In [41] Percival suggests several changes to CPU designs such as cache eviction strategies which respect threading and avoidance of cache sharing between processor cores.

The second approach is to utilize methods which prevent or control the execution of untrusted code. Prevention of untrusted code execution can be achieved by using Trusted Computing concepts in combination with a TPM. Controlling code execution can be achieved by using a technology called *Distributed Information Flow Control* (DIFC). In [42] Krohn et al. introduce an implementation of DIFC called Flume which can run as an extension on Linux and FreeBSD. It promises that it has a smaller impact on existing software and the operating system than for example Asbestos [43] and HiStar [44].

7.5. Discussion

In this chapter several technologies which can be used to improve the segregation of data of different tenants in cloud computing environments are discussed. Because data needs to be protected at all times the three states of data model is used to categorize the different technologies. In the next three subsections the described technologies will be discussed.

7.5.1. Data at Rest

Data segregation in databases is a matter of design decisions. If a very high level trust in segregation is needed one has to choose for separate databases. Of course, this approach comes with a price because a CSP needs more resources to provide a separate database to each customer. If data segregation is still important but the price of separate databases is too high it is possible to choose for separate schemas or even a shared schema. However the latter approach promises to be the cheapest approach it is also the approach that requires most development effort. This is because the segregation controls are part of the software which is provided by the CSP and not part of the database software which was developed by large and specialist database vendors [39].

Segregation at the file system level is mainly a case of access controls of the file system and cloud service. The effectiveness can be increased by using file encryption. While it is most effective if the cryptographic keys are not stored in the cloud this reduces the potential value of cloud services because a cloud service cannot process the information. Recent developments in the field of homomorphic encryption promise the ability to process information in an encrypted state. Unfortunately for many applications it is not yet feasible to use homomorphic encryption on a large scale [45].

7.5.2. Data in Motion

While there is a very strong trend towards connecting every device to the internet and other devices not every network resource needs to be accessible. Especially in enterprise cloud computing environments where data of different companies is transferred between large pools of servers it is important to keep the data streams segregated.

Infrastructure as a service gives customers relatively low level access to cloud infrastructure and thus also to the internal network. VLAN technology can be used to offer secure virtual networks to separate customers. By design the maximum amount of VLANs in a single network is limited to 4k, which is a very limiting factor for the implementation of VLAN technology. Most CSPs have more customers than the maximum number of VLANs. One option is to place small numbers of customers in the same VLAN to reduce the impact of a security breach. Another option which was introduced by Hao et al. [40] separates the network of the CSP in different edge domains which each can contain 4k VLANs. Communication between edge domains is facilitated by FEs which are special network switches which direct network traffic to a specific VLAN. Unfortunately this approach is not very easy to implement. The FEs have to be developed and tested and the network needs to be restructured at a very fundamental level.

For other delivery models than IaaS customers do not have low level access to the network and resources are shared among a larger number of customers. Therefore it is not feasible to create a VLAN for each customer, and thus segregation needs to come from another method. Like in many other cases it is possible to use encryption methods. If data is encrypted while it is transferred it is less likely that a security problem causes data leakage between different customers.

Encryption can be implemented by encrypting and decrypting the data at the application level and transmitting the data over a normal non-secure network. Another approach is to use the same technology that consumers use to connect to a banking application in a secure way: SSL or TLS. These technologies can also be used to create a secure and encrypted connection between services or devices in the infrastructure of a CSP.

7.5.3. Data in Use

CPUs were not specifically designed for multi-tenant use. Standard process separation methods like protected memory spaces and separation of trusted and untrusted programs are still quite effective in keeping the processes and data from different customers separate. Unfortunately we have seen in the paper of Percival [41] that it is possible to take advantage of the processor's cache and its cache eviction strategies. Using this method it is possible to learn about the behavior and possibly the stored data of other processes. The paper demonstrated that certain cryptographic algorithms have predictable behavioral characteristics which can be used to extract cryptographic keys. To mitigate this risk Percival suggests changes to processor designs, operating systems and cryptographic libraries.

Prevention of the execution of programs that eavesdrop on the data and behavior of other programs is another approach which can be used to prevent data segregation incidents at the data in use level. Trusted computing and distributed information flow control promise to control and secure the execution of untrusted code. These methods require specific hardware parts and software, which makes them hard and expensive to implement. Because of the extra computations that are needed the overhead of these methods will also be non-negligible.

7.6. Conclusions

Data segregation is a very important aspect to get right for CSPs. Customers have to trust that their data is safe and will not leak to their competitors or the rest of the world. Because it is so important CSPs invest a lot of effort and money in proper data segregation controls and data leakage prevention. The result is that data segregation incidents are quite rare.

Effective data segregation can only be achieved if data is protected at all levels: data at rest, data in motion and data in use. Especially with IaaS – where users have relative low level access – effort is needed to protect data. Low level access also increases the possibilities to attack and abuse the cloud infrastructure.

Table 10. The characteristics of the discussed data segregation methods.

	Effectiveness	CSP Effort	Customer effort	Impl.	Control	Exists
Separate databases	High. Segregation managed by database software.	Medium. Databases per server limited.	None.	CSP	CSP	Yes
Separate schemas	High. Segregation managed by database software.	Medium. Schemas per databases limited. Restoring backups is difficult.	None.	CSP	CSP	Yes
Shared schema	High if cloud application has good segregation controls.	High. CSP responsible for segregation controls. Restoring backups is hard.	None.	CSP	CSP	Yes
Auth.	High if users and permissions are configured correctly.	Low. File systems support auth.	Medium. User has to authenticate. Customer needs to manage rights.	CSP	Customer / CSP	Yes
Encryption	High if the customer manages the keys.	Low.	Medium. Customer needs to manage keys.	CSP / Customer	Customer / CSP	Yes
VLAN	High. Network traffic of different customers completely isolated.	High. High impact on network infrastructure	None.	CSP	CSP	Yes
SSL/TLS	Medium.	Medium. Software needs to support encrypted connections.	None.	CSP	CSP	Yes

8. Discussion and Conclusions

8.1. Findings

The objective of this thesis is to determine which data privacy related risks of cloud computing are most important and discuss and find technology based methods that can be used to reduce those risks. Finding out which risks are the most important risks was done by interviewing cloud computing consultants. Before the interviews could start an overview of the risk landscape of cloud computing was needed as a basis for the interviews. Therefore the first research question was answered.

> RQ1. What are the (data privacy) risks of cloud computing?

The report from ENISA [2] provided a broad overview of the risks of cloud computing (see Table 2 on page 22). Because other literature did not discuss risks that were not covered by the ENISA report, it could be concluded that it is a sufficiently complete overview. Therefore the risk overview of the ENISA report is used as the answer to RQ1.

Using the reports from ENISA [2], CSA [14] and Gartner [11] an overview of the top risks according to these reports was created (see Table 4 on page 24). This overview was used as a guidance document to structure the interviews. At the beginning of an interview the interviewees were presented with this overview, and they were asked to give their answer to the second research question.

> RQ2. What are the biggest (data privacy) risks of cloud computing?

In Table 5 on page 24 the quantitative results of the interviews are shown as a table with the number of times a risk was classified as one of the most important risks. Every interviewee indicated that regulatory compliance is a very important source of worries for their customers. Especially the European privacy laws cause data location issues and fear of data leakage. To comply with the laws on privacy and governance many privacy and security requirements must be met.

One can see that the interviewees were allowed to choose risks that are not directly data privacy related. For example, vendor lock-in was mentioned during one interview. This was done because for some risks it is not immediately clear if it is a data privacy related risk.

Based on the results of the interviews four different risks were chosen for further research.

First, data location because the results indicate that it is a big problem with many uncertainties. What are the requirements from privacy laws and directives (see [12] and [46])? Does the US government access to data if it is stored there?

The second risk is data deletion. Improper data deletion will increase the impact of data leakage incidents. Failing to delete data when retention periods expire can cause serious compliance issues. With classical IT models it was possible to wipe or destroy hard disks to make sure data is really gone, but with cloud computing – where multi-tenancy is one of the defining characteristics – these methods cannot be used.

The third risk is data leakage with a focus on data leakage which is caused by (malicious) insiders. The Data Loss Database shows that there are many data leakage incidents. Data leakage prevention does not get easier with cloud computing because there is lessening control over the end points that are used to consume the cloud services.

The fourth and last risk is data segregation which is an important factor in data leakage prevention and privacy protection. The multi-tenant nature of cloud computing required the development of effective methods to ensure that data from different customers is separated. Furthermore, not all hardware that is used in cloud computing infrastructure is specifically designed for multi-tenant use and security problems may occur.

RQ3. What are the possible (and adequate) technologies for reducing this risk?

a. Are there adequate existing technologies which are used to reduce this risk?
b. Are there existing technologies from outside the cloud computing domain which can be translated to a technology which can be used to adequately reduce the risk?
c. If a and b do not yield adequate technologies: Is it feasible to find new solutions? And, what are these solutions?

Large parts of RQ3 and its sub-questions have already been answered in the discussion sections of the previous four chapters. This part of the findings will focus on general trends and characteristics and new methods that are introduced in this thesis.

Crypto, crypto and crypto

Using cryptography seems to be a solution for all risks. In every chapter one or more of the discussed technologies involve cryptography, but how effective are these methods? Is it necessary to use it together with other technologies?

One of the most important categories of methods to reduce data location risks is encrypting data before it is stored in the cloud. Personal data can be stored in other countries if that data is encrypted and the keys are not exported to those countries. Because simply encrypting all data is only suitable for cloud storage services and more intelligent services become impossible more clever cryptographic schemes are needed. Schemes that can selectively give access to specific persons, servers or agents in specific regions bring back the possibility to process data while the data is still encrypted. CP-ABE and broadcast encryption are methods that can be used to achieve this. If the cryptographic keys are managed in an effective and secure way these methods can be very effective.

While the classical data deletion methods – disk wiping and destruction – cannot be used to delete data from a specific customer when it is needed crypto-shredding can be used to effectively delete specific data. This method also depends on effective and secure key management.

In the chapter on data segregation we have seen that for all data states cryptography can be used to increase reliability. File systems can be encrypted and data can be transferred using secured TLS connections. And again, the effectiveness of the methods depends on the effectiveness of the key management processes.

Successful cryptography is not only technology, it is also process and people. If keys are managed or used in a poor manner encryption schemes are useless. Therefore it is important to realize that it is not possible to throw in some encryption technology and believe that you are done.

A data provenance based method for data deletion
In section 5.3 on page 38 a new method for data deletion is introduced. Cloud services that keep track of provenance metadata [30][31] know for data artifacts from which other data artifacts it was created. If that relation is reversed one can find out which artifacts were derived from a certain artifact. Thus if that artifact is deleted it is possible to find out which other artifacts also need to be deleted.

The introduced algorithm uses the provenance data to apply operations on the 'offspring' of an artifact. Whether the operation is deletion, anonymization, data masking or something else is determined by a parameter which is added to each provenance edge. The execution of a specific action against a specific data type can be implemented in an adapter program.

If this method is used in a consistent way it is possible to find all copies of data that needs to be deleted. The deletion algorithm will make sure that these copies will be deleted. This will make it less likely that incidents like the Dutch Railways incident which was described in the introduction of chapter 5 will happen. It becomes more transparent where data is in the cloud and data deletion is verifiable if the provenance graph is compared with the log output of the algorithm.

The downside of this approach is that it is necessary to implement data provenance tracking in all parts of the cloud infrastructure. If this is not done properly copies of data artifacts can disappear from the provenance data. This will not be trivial and the amount of metadata that needs to be stored is also significant.

Broadcast Encryption
Broadcast Encryption is an existing method from the digital television domain which can be used to selectively provide access to specific users or devices to encrypted data. In cloud computing it can be used to limit access to the data owner and cloud infrastructure in regions of choice. Because control is largely in the hands of the data owner than CP-ABE and this method is potentially a simpler one it can be an alternative to CP-ABE.

8.2. Reflection and limitations

It is really difficult to validate cloud computing technologies because it is not possible to test a technology in a 'real' cloud. Validating the effectiveness of existing technologies is also difficult because although many CSPs have trust websites where they explain their security controls and efforts, these websites and other official sources from CSPs do not discuss the precise technologies that are used to reduce risks. This makes it complicated to validate the results of this research, and its validity has to come from literature study.

The discussion of existing methods that are used or can be adapted for cloud computing use will not be complete. This is because a wide range of sources was needed to perform this research and it is not possible to get complete knowledge about all the different technologies that exist and can potentially be used to reduce a specific risk.

The risks that are covered in this thesis are – besides data privacy related risks – risks that come forth out of the loss of control risk. Moving to the cloud means that your data is also moved to a location where there is no direct control over it. With classical IT models it was possible to destroy a hard disk if necessary; with cloud computing data from different customers is stored on a single disk and data is replicated across several data centers. The lack of endpoint protection and the rise of BYOD make detecting and preventing data leakage harder. And storing data from different customers in the same database or on the same file system requires trust in the CSPs segregation controls. Trust in the controls of CSPs is the only way in which a company can keep (a feeling of) control over their IT when services are moved to the cloud.

8.3. Contribution

The most important contribution of this thesis consists of an overview of technology based risk reduction methods for data location, deletion, leakage and segregation. This thesis also gives insight into which risks of cloud computing are most important. During the research into the different risks a new application of an existing method – broadcast encryption – and a novel method which uses provenance metadata for effective data deletion are introduced.

List of Figures

Figure 1. The Google Trends report for the query “cloud computing”. 9
Figure 2. Worldwide IT spending by consumption model (from [1]). 9
Figure 3. Outline of the thesis. 11
Figure 4. The delivery models of cloud computing. 14
Figure 5. The responsibilities of the CSP and customer for different delivery models. 14
Figure 6. Countries with an adequate level of protection. EU member states are dark blue, fully recognized countries are light blue and the US is green. 26
Figure 7. In (1) the data subject (*id = 1*) defines that he and a region with *id = 2* may decrypt the data. In (2) a server in the region with *id = 2* successfully decrypts the data, while in (3) the decryption fails because the server is in the region with *id = 3*. 29
Figure 8. Usage of CP-ABE for file encryption. In (1) a private key is generated for a server in Europe, and in (2) for a server in the USA. The data subject defines the region and encrypts his data in (3). In (4) the European server tries to decrypt the file and succeeds, while the USA based server cannot decrypt the file (5). 29
Figure 9. The string ‘Alice’ passes through an encryption proxy before it is transferred over the internet to a cloud environment. When the data is retrieved the cloud responds with the encrypted data and the on premise proxy decrypts the data again. 30
Figure 10. Finding the location of a server by triangulation. The dots represent the probing nodes, the dotted circles represent the distance between the probing node and the server. The place where the circles intersect is the probable location of the server. (Figure from [26]) 31
Figure 11. OPM Edges 37
Figure 12. OPM Nodes 37
Figure 13. Example of an OPM graph (from [30]) 38
Figure 14. `GetTree(A)` retrieves the provenance graph with root artifact `A`. 39
Figure 15. `GetTree(A, G)` adds nodes and edges to the provenance graph for the sub graph which is rooted at `A`. 39
Figure 16. `DeleteArtifact(A)`, the top level method of the algorithm which starts the graph traversal and deletes artifact `A`. 40
Figure 17. `ProcessEdge(E, log)` handles the artifact `E.dest` and recursively processes the part of the graph which originates from `E.dest`. 40
Figure 18. Examples of polymorphic artifact processing methods. 41
Figure 19. The result of running `GetTree("A1")`. This structure corresponds to the graph in Figure 19 (left). 42
Figure 20. The log items that are returned by running `DeleteArtifact("A1")`. 42
Figure 21. A provenance with six artifacts and corresponding provenance edges. The result of deleting artifact A1 is shown in the right provenance graph. 42
Figure 22. The resulting graph after the execution of `DeleteArtifact("A1")`. 43
Figure 23. Incidents by vector (source: www.datalossdb.org) 45
Figure 24. Data segregation based on separate databases. 52
Figure 25. Data segregation based on separate schemas within the same database 52
Figure 26. If `Operation()` and `CryptoOperation()` are homomorphic `CryptoOperation()` can be applied to encrypted data to perform the same computation as `Operation()` and produce the same result in an encrypted way. 53
Figure 27. Data segregation based on a shared schema. 53
Figure 28. The architecture from [40]. 54

List of Tables

Table 1. Levels of risk which were used in the ENISA study (from [2]). ... 22
Table 2. The risks of cloud computing according to ENISA. ... 22
Table 3. Risk distribution ... 23
Table 4. The top risks of cloud computing according to ENISA, CSA and Gartner. ... 24
Table 5. The risks that were mentioned as being very important and the number of times they were mentioned. ... 24
Table 6. The methods to enforce or monitor data location and their characteristics. ... 35
Table 7. The characteristics of the discussed data deletion methods. ... 46
Table 8. Examples of data leakage scenarios with internal and external adversaries which leak data intentionally or by accident. ... 47
Table 9. The characteristics of the discussed data leakage prevention and detection methods. ... 52
Table 10. The characteristics of the discussed data segregation methods. ... 59

List of Abbreviations

API	Application Programming Interface
BYOD	Bring Your Own Device
CP-ABE	Ciphertext Policy - Attribute Based Encryption
CPU	Central Processing Unit
CSP	Cloud Service Provider
DAG	Directed Acyclic Graph
DaR	Data at Rest
DIFC	Distributed Information Flow Control
DiM	Data in Motion
DiU	Data in Use
Dutch DPA	Dutch Data Protection Agency (College Bescherming Persoonsgegevens)
ENISA	European Network and Information Security Agency
E-P3P	Platform for Enterprise Privacy Practices
EU	European Union
FE	Forwarding Element
HE	Homomorphic Encryption
HTTP	HyperText Transfer Protocol
IaaS	Infrastructure as a Service
IPSec	Internet Protocol Security
IT	Information Technology
MLAT	Multilateral Legal Assistance Treaty
OPM	Open Provenance Model
PaaS	Platform as a Service
PII	Personal Identifiable Information
REST	Representational State Transfer
SaaS	Software as a Service
SOAP	Simple Object Access Protocol
SSH	Secure Shell
SWOT	Strengths, Weaknesses, Opportunities and Threats
TLS	Transport Layer Security
TPM	Trusted Platform Module
US	United States
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPN	Virtual Private Network

Bibliography

[1] IDC, "IT Cloud Services Forecast: 2009-2013," 2009. [Online]. Available: http://blogs.idc.com/ie/?p=543. [Accessed: 21-Sep-2012].

[2] ENISA, "Cloud Computing: Benefits, Risks and Recommendations for Information Security," 2009.

[3] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, "Controlling Data in the Cloud : Outsourcing Computation without Outsourcing Control," in *Proceedings of the 2009 ACM workshop on Cloud computing security*, ACM, 2009, pp. 85–90.

[4] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud security and privacy: an enterprise perspective on risks and compliance*. O'Reilly Media, Inc., 2009.

[5] L. M. Vaquero, L. Rodero-merino, J. Caceres, and M. Lindner, "A Break in the Clouds : Towards a Cloud Definition," *Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2009.

[6] NIST, "SP800-145: The NIST Definition of Cloud Computing." [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. [Accessed: 21-Sep-2012].

[7] ISO, "ISO 31000:2009 - Risk management - Principles and guidelines." [Online]. Available: http://www.iso.org/iso/catalogue_detail?csnumber=43170.

[8] Institute of Management Accountants, "Enterprise Risk Management: Tools and Techniques for Effective Implementation." [Online]. Available: http://poole.ncsu.edu/erm/documents/IMAToolsTechniquesMay07.pdf. [Accessed: 21-Sep-2012].

[9] Microsoft Corp., "A Guide to Data Governance for Privacy, Confidentiality, and Compliance. Part 3: Managing Technological Risk," 2010.

[10] CSA, "Top Threats to Cloud Computing," 2010. [Online]. Available: http://www.cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf. [Accessed: 21-Sep-2012].

[11] Gartner, "Assessing the Security Risks of Cloud Computing," 2008.

[12] "Data Protection Directive (Directive 95/46/EC)," 1995. [Online]. Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML. [Accessed: 21-Sep-2012].

[13] European Commission, "Commission decisions on the adequacy of the protection of personal data in third countries." [Online]. Available: http://ec.europa.eu/justice/data-protection/document/international-transfers/adequacy/index_en.htm#h2-5. [Accessed: 21-Sep-2012].

[14] Cloud Security Alliance, “Security guidance for critical areas of focus in cloud computing.” [Online]. Available: https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf. [Accessed: 21-Sep-2012].

[15] R. Gellman, “Privacy in the Clouds : Risks to Privacy and Confidentiality from Cloud,” in *World Privacy Forum*, 2009, pp. 1–26.

[16] “USA PATRIOT Act (H.R. 3162).” [Online]. Available: http://epic.org/privacy/terrorism/hr3162.html. [Accessed: 21-Sep-2012].

[17] W. Maxwell and C. Wolf, “A Global Reality : Governmental Access to Data in the Cloud,” 2012. [Online]. Available: http://www.hldataprotection.com/uploads/file/Hogan Lovells White Paper Government Access to Cloud Data Paper (1).pdf. [Accessed: 21-Sep-2012].

[18] “Geo Location Enables Developers To Choose Data Centers and Group Applications & Storage.” [Online]. Available: http://blogs.msdn.com/b/windowsazure/archive/2009/03/18/geo-location-enables-developers-to-choose-data-centers-and-group-applications-storage.aspx. [Accessed: 21-Sep-2012].

[19] SC Magazine, “Google: Who cares where your data is?” [Online]. Available: http://www.scmagazine.com.au/News/260041,google-who-cares-where-your-data-is.aspx. [Accessed: 21-Sep-2012].

[20] J. Noltes, “Data location compliance in cloud computing,” 2011. [Online]. Available: http://wwwhome.cs.utwente.nl/~franqueirav/Master/Master_Thesis_report_JNoltes.pdf. [Accessed: 21-Sep-2012].

[21] G. Karjoth, M. Schunter, and M. Waidner, “Platform for enterprise privacy practices: Privacy-enabled management of customer data,” in *Privacy Enhancing Technologies*, Springer, 2003, pp. 194–198.

[22] Q. Tang, “On using encryption techniques to enhance sticky policies enforcement,” 2008.

[23] D. Boneh, C. Gentry, and B. Waters, “Collusion Resistant Broadcast Encryption With Short Ciphertexts and Private Keys,” in *Advances in Cryptology - CRYPTO 2005*, 2005, pp. 258–275.

[24] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-Policy Attribute-Based Encryption,” in *2007 IEEE Symposium on Security and Privacy (SP ’07)*, 2007, pp. 321–334.

[25] “Database.com Data Residency Option.” [Online]. Available: http://www.salesforce.com/dro/. [Accessed: 21-Sep-2012].

[26] Z. N. J. Peterson, M. Gondree, and R. Beverly, “A Position Paper on Data Sovereignty : The Importance of Geolocating Data in the Cloud,” in Proceedings of the 8th USENIX conference on Networked systems design and implementation, 2011.

[27] J. Ruiter, “The Relationship between Privacy and Information Security in Cloud Computing Technologies,” Vrije Universiteit, 2009.

[28] P. Gutmann, “Secure Deletion of Data from Magnetic and Solid-State Memory,” in *Proceedings of the 6th conference on USENIX Security Symposium, Focusing on Applications of Cryptography*, 1996, vol. 6.

[29] “Cloud Data Security: Archive and Delete.” [Online]. Available: https://securosis.com/blog/cloud-data-security-archive-and-delete-rough-cut. [Accessed: 21-Sep-2012].

[30] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, and P. Groth, “The Open Provenance Model,” 2010. [Online]. Available: http://eprints.soton.ac.uk/264979/1/opm.pdf. [Accessed: 21-Sep-2012].

[31] K.-K. Muniswamy-Reddy and M. Seltzer, “Provenance as first class cloud data,” *ACM SIGOPS Operating Systems Review*, vol. 43, no. 4, p. 11, Jan. 2010.

[32] SANS Institute, “InfoSec Reading Room ho ll r igh.” [Online]. Available: http://www.sans.org/reading_room/whitepapers/awareness/data-leakage-threats-mitigation_1931. [Accessed: 21-Sep-2012].

[33] PCI Security Standards Council, “Payment Card Industry (PCI) Data Security Standard,” 2010. [Online]. Available: https://www.pcisecuritystandards.org/documents/pci_dss_v2.pdf. [Accessed: 21-Sep-2012].

[34] M. Alawneh and I. M. Abbadi, “Sharing but Protecting Content Against Internal Leakage for Organisations,” *Data and Applications Security*, vol. XXII, pp. 238–253, 2008.

[35] M. Alawneh and I. M. Abbadi, “Preventing information leakage between collaborating organisations,” *Proceedings of the 10th international conference on Electronic commerce - ICEC ’08*, p. 1, 2008.

[36] S. J. Stolfo and M. B. Salem, “Fog Computing : Mitigating Insider Data Theft Attacks in the Cloud Position Paper,” *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, pp. 125–128, 2012.

[37] T. Sasaki, “A Framework for Detecting Insider Threats using Psychological Triggers,” *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 3, no. 1/2, pp. 99–119, 2012.

[38] Information Systems Audit and Control Association (ISACA), “Data Leak Prevention,” 2010. [Online]. Available: http://www.isaca.org/Knowledge-Center/Research/Documents/Forms/DispForm.aspx?ID=6909. [Accessed: 21-Sep-2012].

[39] Microsoft, “Multi-Tenant Data Architecture.” [Online]. Available: http://msdn.microsoft.com/en-us/library/aa479086.aspx. [Accessed: 21-Sep-2012].

[40] F. Hao, T. V. Lakshman, S. Mukherjee, H. Song, and B. Labs, “Secure Cloud Computing with a Virtualized Network Infrastructure,” *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, 2010.

[41] C. Percival, “Cache missing for fun and profit,” BSDCan 2005, 2005.

[42] M. Krohn, A. Yip, M. Brodsky, N. Cliffer, M. F. Kaashoek, E. Kohler, and R. Morris, “Information flow control for standard OS abstractions,” *ACM SIGOPS Operating Systems Review*, vol. 41, no. 6, p. 321, Oct. 2007.

[43] S. Vandebogart, P. Efstathopoulos, E. Kohler, M. Krohn, C. Frey, D. Ziegler, F. Kaashoek, R. Morris, and D. Mazières, “Labels and event processes in the Asbestos operating system,” *ACM Transactions on Computer Systems,* vol. 25, no. 4, p. 11–es, Dec. 2007.

[44] N. Zeldovich, S. Boyd-wickizer, E. Kohler, and D. Mazi, “Making Information Flow Explicit in HiStar,” *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation*, vol. 7, pp. 263–278, 2006.

[45] A. Basu, “Practical privacy using homomorphic encryption – a myth or reality ? The magic of homomorphic encryption,” 2012. [Online]. Available: https://cloudsecurityalliance.org/wp-content/uploads/2012/05/Day1_0930_Track2_Session_BasuAnirban_-_Cloud1_securecloud2012.pdf. [Accessed: 21-Sep-2012].

[46] “Wet Bescherming Persoonsgegevens.” [Online]. Available: http://wetten.overheid.nl/BWBR0011468. [Accessed: 21-Sep-2012].

Appendices

Appendix A. Deletion algorithm

provdel.py

```
import networkx as nx

class ProvDel:

  graph = nx.DiGraph()

  def Dump(self):
    print(self.graph.nodes())
    print(self.graph.edges())

  def GetGraph(self):
    return self.graph

  def CreateArtifact(self, name, data):
    return self.graph.add_node(name, value=data)

  def DeriveArtifact(self, name, parents):
    derivedValue = name + “, “
    for p in parents:
      derivedValue += self.graph.node[p[0]][‘value’] + “, “

    self.CreateArtifact(name, derivedValue)


    for p in parents:
      self.graph.add_edge(name, p[0], action=p[1])

  def GetTree(self, root):
    print “Node(“ + root + “) = \”” + self.graph.node[root][‘value’] + “\””
    edges = self.GetProvenanceEdges(root)
    for e in edges:
      print “Edge(“ + e[0] + “, “ + e[1] + “) with action “  + self.graph.edge[e[0]]
[e[1]][‘action’]
    self.GetTree(e[0])


  def GetProvenanceEdges(self, name):
    res = list()
    for e in self.graph.edges():
      if e[1] == name:
        res.append(e)
          return res

  def DeleteArtifact(self, name):
    log = list()
    edges = self.GetProvenanceEdges(name)
    for e in edges:
      log += self.ProcessEdge(e, name)

    log.append(self.Delete(name))
    return log
```

```
def ProcessEdge(self, edge, root):
    log = list()
    action = self.graph.edge[edge[0]][edge[1]][‘action’]
    if action != “none”:
      edges = self.GetProvenanceEdges(edge[0])
      for e in edges:
        log += self.ProcessEdge(e, root)

    if action == “delete”:
      log.append(self.Delete(edge[0]))
    elif action == “partial_delete”:
      log.append(self.PartialDelete(edge[0], root))
    elif action == “anonymize”:
      log.append(self.Anonymize(edge[0], root))

    return log

  def Delete(self, name):
    self.graph.remove_node(name)
    return “Deleted artifact “ + name

  def PartialDelete(self, art, root):
    delValue = self.graph.node[root][‘value’]
   self.graph.node[art][‘value’] =     self.graph.node[art][‘value’].replace(delValue,
“”)
    return “Partially deleted artifact “ + art

  def Anonymize(self, art, root):
    delValue = self.graph.node[root][‘value’]
     self.graph.node[art][‘value’] = self.graph.node[art][‘value’].replace(delValue,
“***”)
    return “Anonymized artifact “ + art
```

test.py

```
import provdel
pd = provdel.ProvDel()

pd.CreateArtifact(“A1”, “A1”)
pd.CreateArtifact(“A2”, “A2”)

pd.DeriveArtifact(“A3”, [(“A1”, “delete”), (“A2”, “delete”)])

pd.DeriveArtifact(“A4”, [(“A3”, “anonymize”)])
pd.DeriveArtifact(“A5”, [(“A3”, “none”)])

pd.DeriveArtifact(“A6”, [(“A5”, “delete”)])

pd.GetTree(“A1”)
print “=====================”
print pd.DeleteArtifact(“A1”)
print “=====================”
pd.Dump()
```