Idea: Optimising Multi-Cloud Deployments with Security Controls as Constraints

Philippe Massonet[2], Jesus Luna[1], Alain Pannetrat[1], Ruben Trapero[3],

[1] Cloud Security Alliance (Europe), United Kingdom
{jluna, apannetrat}@cloudsecurityalliance.org
[2]Centre d'Excellence en Technologies de l'Information et de la Communication, Belgium
philippe.massonet@cetic.be
[3]Department of Computer Science, Technische Universitat Darmstadt, Germany
rtrapero@deeds.informatik.tu-darmstadt.de

Abstract. The increasing number of cloud service providers (CSP) is creating opportunities for multi-cloud deployments, where components are deployed across different CSP, instead of within a single CSP. Selecting the right set of CSP for a deployment then becomes a key step in the deployment process. This paper argues that deployment should take security into account when selecting CSP. This paper makes two contributions in this direction. First the paper describes how industrial standard security control frameworks may be integrated into the deployment process to select CSP that provide sufficient levels of security. It also argues that ability to monitor CSP security should also be considered. The paper then describes how security requirements may be modelled as constraints on deployment objectives to find optimal deployment plans. The importance of using cloud security standards as a basis for reasoning on required and provided security features is discussed.

Keywords: Cloud, Security, Deployment, Optimization, Security controls, Security service level agreements, Monitoring

1 Introduction

The growing number of CSP offering infrastructure services (IaaS) opens up opportunities for benefitting from the advantages of deploying applications over multiple CSP. Multi-cloud systems (MCS) [8, 9] involve deploying components of a single application on more than one CSP. There are multiple reasons justifying MCS and they range from improving fault tolerance, to minimizing cost of deployment, or to improving response time by deploying components closer to customer locations.

However moving to MCS raises new challenges such as being able to deploy, undeploy and redeploy easily from one CSP to another. The general approach taken in this paper is to build a deployment model that is independent of any specific CSP [7]. A deployment process then transforms the CSP independent deployment model into an executable deployment plan for a specific CSP. Such a model must capture functional as well as non-functional deployment requirements. In this paper we suggest to take security into account by expressing security requirements in terms of cloud security control frameworks such as CCM [1]. Such frameworks provide some degree of security assurance and transparency, because auditors evaluate the security of a cloud service against a set of “controls” chosen from a reference “security control framework”. Security controls remain high level and can be implemented in many different ways.

Workgroups at the European Network and Information Security Agency (ENISA) and the National Institute of Standards and Technology (NIST) have identified [3, 4] that specifying security service level objectives (SLO) in security Service Level Agreements (secSLA) is a useful tool to establish common semantics supporting the description of security assurances. In order to present the key elements driving the adoption of useful SLO to select cloud providers, this paper explores (i) how security controls and security level objectives can be modelled as constraints on a cloud deployment, and (ii) how these security constraints can be used to select the best set of cloud providers on which to deploy the different components of a MCS. The paper discusses the use of SLO as a basis for continuous monitoring of cloud service security as future work at the end of the paper. The arguments presented in this paper are the result of our research and field experience in relevant academic/industrial projects (e.g., EU funded SPECS [5], CUMULUS [6] and PAASAGE [7]), standardization bodies (e.g., NIST, ETSI and ISO/IEC), and related Cloud Security Alliance workgroups (e.g., Cloud Trust Protocol –CTP- and Service Level Agreements).

This paper is organized as follows: Section 2 describes the concept of multi-cloud and identifies challenges for deployment. Section 3 argues that industry standards for security controls and security level objectives should be used to allow comparing side-by-side cloud providers. Section 4 describes how selecting the best set of cloud providers for a cloud deployment can be modelled as a combinatorial optimization problem, and that security controls and SLO can be modelled as constraints. The section concludes with some preliminary experimental results.

2 Multi-Cloud Applications and Case Study

Figure 1 illustrates a MCS where a cloud application is deployed both in private and public clouds in different countries and jurisdictions. The objective is to locate the web server close to customers in order to reduce response time as much as possible on mobile devices such as smartphones and tablets. The database server (DBS) is kept in a private cloud in Spain at the head office of the company while the web servers (WS) and application servers (AS) are deployed in public clouds located close to customers in the United Kingdom and Germany.

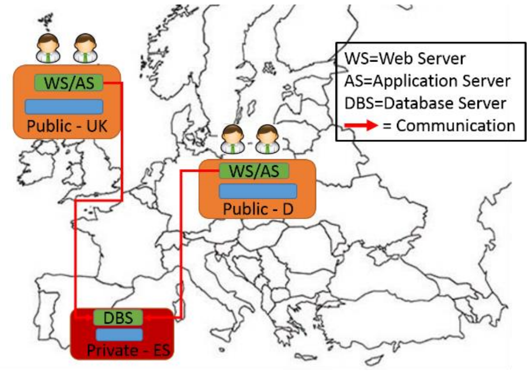


Figure 1 A multi-cloud deployment in Europe

In this example, it is realistic to suppose that each one of the CSPs involved has implemented its own security mechanisms, controls and policies. Under these circumstances the customer might be also faced with the following security assurance challenges: How to select the right public CSP, based on the component's security requirements? How to continuously monitor the overall cloud infrastructure to assess that all security requirements are fulfilled? Because each provider in the MCS can implement their security controls in a different way, what is the aggregated/overall security assurance level provided to the customer?

The provision of security assurance to the cloud customer in the presented MCS scenario covers many challenges such as cloud provider selection, continuous security monitoring or aggregation of security levels. This paper focuses on the first challenge i.e., selecting the right set of CSP's so that security requirements are satisfied when deploying the application.

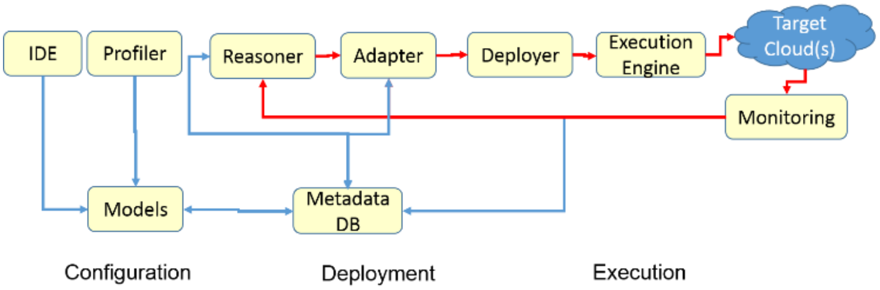
3 Model-Based Deployment of Multi-Cloud Applications

This section introduces the context of this paper, namely that deployment should be based on models, and that deployment models should be optimised with respect to objectives and requirements.

3.1 Model-based Deployment Workflow

This section briefly describes the different phases of the deployment workflow and components [7]. The figure below shows three workflow phases: configuration, deployment and execution. The configuration phase involves building a model of the application components to be deployed. This involves describing in a model the artifacts to be deployed, the communication links between artifacts, the scalability requirements of each artifact, … The model includes a description of security requirements for each deployable artefact. In the deployment phase the deployment model is analysed by a component called the "Reasoner" to produce an optimal deployment plan that meets deployment objective and constraints defined in the model. In terms of security the

“Reasoner” component will compare and match security requirements to CSP features. In the execution phase of the workflow the “Adapter”, “Deployer” and “Execution engine” execute the deployment plan resulting in a multi cloud deployment that is monitored by the “Adapter”. The “Adapter” controls the run-time feedback loop by analysing the monitoring data and performs run-time adaptations. From the security point of view monitoring data about security controls is analysed. If model violations cannot be solved at run-time, then control is passed to a design-time feedback loop where the MCS is stopped and the “Reasoner” calculates a new deployment plan that solves the model violations.



The rest of the paper focuses on the utility function that is used by the “Reasoner” to find an optimal deployment plan that includes security constraints.

3.2 Modelling Deployment Security Requirements

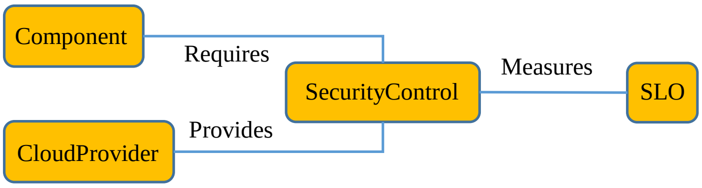


Figure 2 Security Control Concept in the Meta-model

The MCS is assumed to be composed of several components, and it is assumed that each component can be deployed separately on cloud resources. Figure 2 shows a fragment of the deployment meta-model that shows how security meta-concepts are related to deployable artifacts. “Components” are deployable artifacts and can “require” a “security control” in order to be deployed. “Security control” can be “provided” by “Cloud Providers”. “Security Controls” are abstract and are difficult to monitor. “SLO” on the other hand are measurable and are related to “SecurityControls”.

From the security perspective these meta-concepts provide the basis for matching required “security controls” and “SLO” with CSP “security controls” and “SLO”. In order to produce an executable deployment plan, a cloud provider must be selected for each application component. To address the security assurance and transparency issues discussed earlier in the paper, most CSPs would submit their service to certification by independent third party auditors, based on well-established standards such as ISO/IEC 27001 [10], PCI-DSS [11], or CCM [1] for example. Security control frameworks can be complemented with security Service Level Agreements (secSLA). This approach is based on the assessment of measurable SLOs in secSLAs.

4 Optimizing Multi-Cloud Deployments with Security Constraints

4.1 A Cloud Deployment Utility function with Security Constraints

As was described in Section 3 the "Reasoner" component analyses the deployment model to build a utility function that optimises the deployment objectives and satisfies all constraints. The objective function may cover multiple criteria such as cost, availability or response time. The objective function and constraints are defined by analysing the deployment model. In this section we illustrate the approach by describing a specific objective function and constraints for the running case study. The utility function is then used to produce a deployment plan that makes trade-offs between security and other constraints.

The objective function shown in line (1) of the table below minimises the total deployment cost, i.e. the sum of the costs of all the application components. In the process it must assign a cloud provider to each component. Line (2) defines the deployment cost of a component as the sum of cost for the virtual machines and the cost of the storage measured in terms of I/O operations. The function "provider(c)" in line (5) returns the cloud provider that has been selected for a given component. Line (3) shows that the application to be deployed is composed of three components: two application/web servers, and one database server. It also defines "P" the list of 5 potential cloud providers for deploying the components. Providers 0 and 1 are private clouds, and the others are public clouds. Line (4) defines some bounds for the total cost of a single deployment. Line (6) shows how an availability constraint may be defined for a given component and defines valid values for availability. Line (7) shows that security controls are modelled as a Boolean choice: they can be required or not for a given component. Requirements on SLO are modelled in the same manner in line (8).

$$(1)\ Min \sum_{comp=1}^{comp\_max} comp\_deployment\_cost_{\mathrm{comp}}$$

$$(2)\ \mathrm{comp\_deployment\_cost(c)} = requiredVM(c) * vm\_cost\big(provider(c)\big) + requiredIO(c) * IO\_cost(provider(c)))$$

$$(3)\ \mathrm{c} \in \mathrm{C},\ \mathrm{C}=\{\mathrm{ws/as},\ \mathrm{ws/as},\ \mathrm{dbs}\},\ \mathrm{P}=\{0,\ \ldots,\ 4\}$$

$$(4)\ comp\_deployment\_cost \in (minCost, maxCost),\ (5)\ provider(c)\ \in P$$

$$(6)\ requiredAvailability(c) \geq a, \qquad where\ 99{,}9 \leq a \leq 99{,}9999$$

$$(7)\ requiredSC(c, sc) = b, where\ b \in \{0{,}1\}, and\ sc\ \in SecurityControls$$

$$(8)\ requiredSLO(c, slo) = b, where\ b \in \{0{,}1\}, and\ slo\ \in SLObjectives$$

Figure 3 Equations for minimising deployment cost and selecting a provider per component

To illustrate how security is modelled Figure 4 shows a partial decomposition of CCM security control AIS-04 into several intermediate security controls, e.g. “Prevent improper disclosure”, and finally into three secSLOs that can be effectively assessed and monitored in a cloud infra-structure (provided they have been documented using a model like NIST [8]). Take for example “Country level anchoring” SLO which is defined as follows: “this attribute indicates that all processing operations applicable to the resource only take place within a set of predefined countries”. The value associated with such SLO is “a vector of strings representing a two-letter ISO-3166-1 country code”. This SLO allows expressing a constraint on the jurisdiction in which a cloud deployment can be made, and to subsequently monitor that the deployment has not moved outside of this jurisdiction.

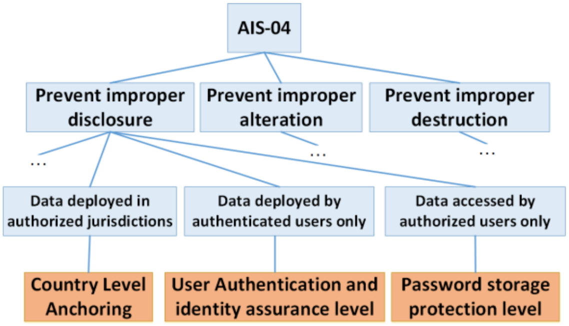
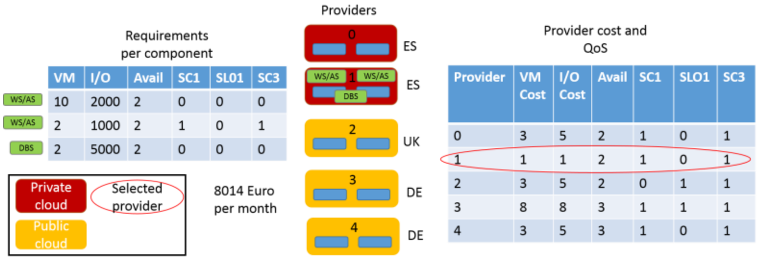

AIS-04
Prevent improper disclosure
Prevent improper alteration
Prevent improper destruction
…
…
…
Data deployed in authorized jurisdictions
Data deployed by authenticated users only
Data accessed by authorized users only
Country Level Anchoring
User Authentication and identity assurance level
Password storage protection level

Figure 4 Decomposition of control AIS-04 into SLOs

To illustrate how the constraints on SC and SLO are instantiated consider the following constraints “ $(a)\ requiredSC(ws/as, AIS04) = 1$, (b) requiredSLO(ws/as, CountryLevelAnchoring, DE)=1”. The first constraint requires that the provider that is selected for deploying the “ws/as” component must have implemented the “AIS-04” security control. “AIS-04” describes general mechanisms for controlling data exchange between jurisdictions. If we want to limit data exchange to a list of specific jurisdictions, then we need to add constraint (b) that requires providers to support SLO “CountyLevelAnchoring” and limit the location of data to Germany (“DE”).

4.2 Preliminary Experimental Results

The above optimisation problem was modelled with a constraint programming solver [12]. This section describes some preliminary experimental results, by showing different security requirements and the corresponding deployment models. In Figure 5 we see a first deployment model produced by the constraint program. The right part of the figure shows the cloud provider attributes. The deployment model that has been produced by the constraint program shows the deployment that minimises total cost of the deployment. As can be seen in Figure 5, provider 1 that is located in Spain has been selected for all three components because it is the least expensive provider that satisfies all constraints. The total deployment cost is 8014 euro per month. In fact all cloud providers satisfy all component constraints except provider 2, because he does not provide the security controls that are required by the second WS/AS component.



Requirements per component

	VM	I/O	Avail	SC1	SL01	SC3
WS/AS	10	2000	2	0	0	0
WS/AS	2	1000	2	1	0	1
DBS	2	5000	2	0	0	0

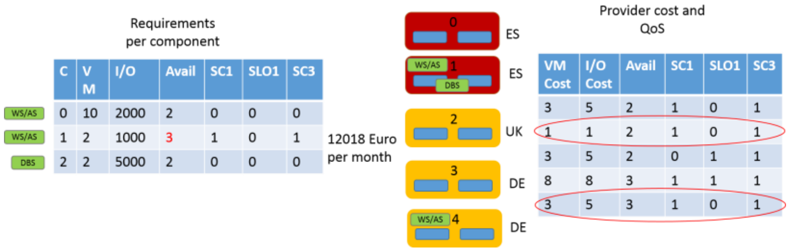
8014 Euro per month

Provider cost and QoS

Provider	VM Cost	I/O Cost	Avail	SC1	SLO1	SC3
0	3	5	2	1	0	1
1	1	1	2	1	0	1
2	3	5	2	0	1	1
3	8	8	3	1	1	1
4	3	5	3	1	0	1

Figure 5 Minimal cost deployment

Figure 6 shows slightly different deployment requirements on the components and the resulting new deployment model. In the table of the left part of the figure, availability for the second WS/AS component has been increased to three nine, i.e. “99,999”. The previous deployment is no longer a solution because provider 1 only offers “99,99” availability. For the second WS/AS component, the only two providers that offer “99,999” availability are Providers 3 and 4 both located in Germany. Provider 4 has been selected because it is less expensive than provider 3. The figure shows the resulting deployment where the second WS/AS component is deployed on provider 4. The other two components are deployed as before on provider 1.



Requirements per component

	C	VM	I/O	Avail	SC1	SLO1	SC3
WS/AS	0	10	2000	2	0	0	0
WS/AS	1	2	1000	3	1	0	1
DBS	2	2	5000	2	0	0	0

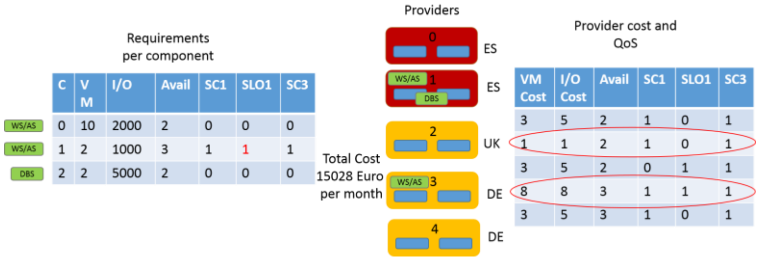
12018 Euro per month

Provider cost and QoS

VM Cost	I/O Cost	Avail	SC1	SLO1	SC3
3	5	2	1	0	1
1	1	2	1	0	1
3	5	2	0	1	1
8	8	3	1	1	1
3	5	3	1	0	1

Figure 6 Deployment for higher availability

Figure 7 shows another change in the deployment requirements and the resulting new deployment model. An SLO, e.g. “requiredSLO(ws/as, CountryLevelAnchoring, DE)=1” is now required for the second WS/AS component. This is shown in red in the table of the left part of the figure. Provider 4 is no longer a solution because it does not offer the required SLO. Even though it is located in Germany it does not provide any data location monitoring data. Provider 3 is the only provider that offers this SLO and is thus the selected provider even though it is the most expensive.



Requirements per component

	C	VM	I/O	Avail	SC1	SLO1	SC3
WS/AS	0	10	2000	2	0	0	0
WS/AS	1	2	1000	3	1	1	1
DBS	2	2	5000	2	0	0	0

Total Cost 15028 Euro per month

Provider cost and QoS

VM Cost	I/O Cost	Avail	SC1	SLO1	SC3
3	5	2	1	0	1
1	1	2	1	0	1
3	5	2	0	1	1
8	8	3	1	1	1
3	5	3	1	0	1

Figure 7 Deployment with extra security

5 Discussion and Conclusions

Selecting could providers for MCS components is a special case of the general problem of workload placement. The main contribution of this paper is to show that security requirements can deploying workloads to multiple cloud providers using industry standard SC and SLO such as CCM [1]. Related efforts for selecting cloud providers [14] have focused on comparing cloud providers by measuring the level of security they provide. Compared to this paper this work focuses on security and does not take into account other types of requirements. Furthermore it requires that every security requirement be measurable. In our approach we do not attempt to measure security levels, since requirements are expressed as constraints. This approach is more flexible when it is difficult to quantity the requirements. In future work security requirements could be integrated into the objective function provided that they can be quantified. Other related research efforts have worked on integrating security into SLAs by describing fine grained security properties in a security property specification language [15]. In this paper we have combined coarse grained SC requirements with finer grainer SLO in the deployment decision making. In future work we could also integrate security properties in the decision making since they have been modelled in the security deployment metamodel. Future work will integrate the utility function into a cloud deployment platform [7], and will investigate how to make adaptations in the run-time feedback loop by analysing SLO monitoring data to solve SLO violations.

Acknowledgements. This work is partially funded by the EU FP7 PaaSage and EU FP7 SPECS projects

References

1. Cloud Control Matrix. Online: http://www.cloudsecurityalliance.org/cm.html. 2011
2. Cloud Security Alliance. The Security, Trust & Assurance Registry (STAR). Online: https://cloudsecurityalliance.org/star/. Last Access: 2014
3. Dekker, M., and G. Hogben. "Survey and analysis of security parameters in cloud SLAs across the European public sector." Available online in http://www. enisa. europa. eu/, 2011.
4. NIST "Cloud Computing: Cloud Service Metrics Description (RATAX)". 2014.
5. SPECS home page: http://specs-project.eu/. Last access: 2014.
6. CUMULUS project home page: http://www.cumulus-project.eu. Last access: 2014.
7. PASSAGE project home page: http://www.passage-project.eu/. Last access: 2014.
8. Cloud computing, http://en.wikipedia.org/wiki/Cloud_computing#Multicloud
9. Multi cloud, http://en.wikipedia.org/wiki/Multicloud
10. Brenner, Joel. "ISO 27001: Risk management and compliance." RISK MANAGEMENT-NEW YORK- 54, no. 1, 2007: 24.
11. Industry, Payment Card. "Data security standard." Requirements and Security Assessment Procedures, Version 3. 2013.
12. Choco Solver, http://www.emn.fr/z-info/choco-solver/
13. NIST "Cloud Computing: Cloud Service Metrics Description (RATAX)". Working document. 2014.
14. Jesus Luna Garcia, Tsvetoslava Vateva-Gurova, Neeraj Suri, Massimiliano Rak, and Loredana Liccardo. "Negotiating and Brokering Cloud Resources based on Security Level Agreements.", CLOSER, page 533-541. SciTePress, (2013)

15. Alain Pannetrat, Giles Hogben et al. “D2.1 Security-aware SLA specification language and Cloud security dependency model”, CUMULUS project deliverable, 2013.