A Broker Framework for Secure and Cost-Effective Business Process Deployment on Multiple Clouds

Elio Goettelmann, Karim Dahman, Benjamin Gateau, Claude Godart

A Broker Framework for Secure and Cost-Effective Business Process Deployment on Multiple Clouds

Elio Goettelmann[1,2], Karim Dahman[3], Benjamin Gateau[2] and Claude Godart[1]

[1] LORIA - INRIA Grand Est, Université de Lorraine, Nancy, France.
[2] CRP Henri Tudor, Kirchberg, Luxembourg.
[3] Blu Age - Netfective Technology, Pessac, France.
elio.goettelmann@tudor.lu, k.dahman@bluage.com, benjamin.gateau@tudor.lu, claude.godart@loria.fr

Abstract. Security risk management on information systems provides security guarantees while controlling costs. But security risk assessments can be very complex, especially in a cloud context where data is distributed over multiple environments. To prevent costs from becoming the only cloud selection factor, while disregarding security, we propose a method for performing multiple cloud security risk assessments. In this paper we present a broker framework for balancing costs against security risks. Our framework selects cloud offers and generates deployment-ready business processes in a multi-cloud environment.

Keywords: Business Process, Security Risk Management, Cloud

1 Introduction

The Cloud business model proposes a multitude of different services, at different prices, and with various quality levels. While the use of cloud computing can reduce costs, the selection of a solution is time consuming. For this purpose, cloud brokers have emerged; they can help cloud consumers to select adequate solutions, by comparing existing offers, essentially against their prices.

But security is still an important factor for a cloud selection process. As cloud computing presents new kinds of security risks ([3], [5]), they need to be treated before wider adoption. Novel methods have to be defined in order to prevent these potential losses on companies.

In turn, distributing software over multiple locations increases the complexity of gathering sensitive business information. In this paper, we propose a framework for cloud brokers which helps them analyze the security levels of different cloud offers, following standard risk assessment methodologies, with respect to cloud offers.

The paper is organized as follows. Section 2 presents a motivating example used to demonstrate the purpose and the scope of our framework. Section 3 describes our tool, its implementation on the motivating example and experimental results. Section 4 and Section 5 discuss respectively related and future work.

2 Motivating Example and Overview

In this section, we introduce a motivating example to illustrate our framework. Then we give an overview of our approach for selecting clouds when deploying business processes.

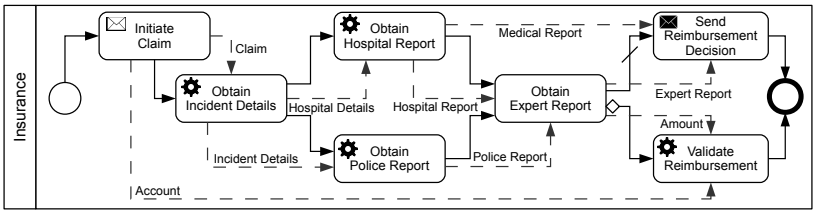

Fig. 1. Business Process Motivation Example: Insurance Claim Recovery Chain.

Consider an *insurance claim recovery chain* [7] as a BPMN business process model depicted in Fig. 1. This process is initiated when an *insurance* company receives a claim recovery declaration from a *beneficiary*. To obtain details about the incident, the *emergency* service is invoked. The *hospital* and *police* reports are required by the *expert* to decide if the *reimbursement* will be accepted or not. If so, the *bank* is requested and a notification is sent to the *beneficiary*.

Now suppose that the insurance company wants to outsource the software supporting this process to the cloud to reduce costs. It has not necessarily the knowledge of moving it effectively on its own. A cloud broker could support the company by **evaluating the security risks** of a cloud outsourcing, **selecting the adequate offers** and **decomposing the process** to deploy it on multiple clouds. These three tasks are detailed below.

Our tool consists in a design-time framework for producing secure and cost-effective business processes on multiple cloud. It is illustrated in Fig.2.

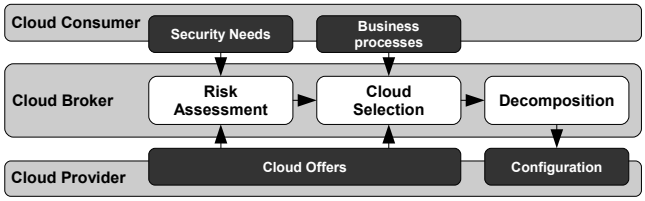

Fig. 2. Our design-time framework for multi-cloud business process deployment.

First, a cloud broker requests the *security needs* as non-functional requirements from the cloud consumer and analyzes the *cloud offers* from the cloud provider to realize a **risk assessment**.

Second, the broker uses the *business processes* from the cloud consumer to **select** the adapted *cloud offers* based on the functional requirements, the costs and the previously calculated risk.

Third, the broker **decomposes** the business process into smaller parts, as each task can be enacted on a different cloud offer. The generated *configuration* is the assignment of these process fragments to cloud offers. The decomposition itself has already been addressed in [8].

3 Implementation, experimentation and evaluation

To demonstrate the feasibility of the approach, we illustrate on our motivating example the three previously defined steps of the broker tool.

3.1 Risk Assessment

The risk assessment is threefold: security needs definition, risk evaluation and cloud provider exclusion.

Security needs definition Our framework uses the five CIANA objectives (*Confidentiality, Integrity, Availability, Non-Repudiation, Authenticity*) to define the *security need* of each data object of the process (Fig. 1). If the data object needs the objective, the value is equal to 1, otherwise to 0 (see Table 1). Typically, these security needs are negotiated by a risk manager with the cloud consumer.

Table 1. Annotations of the security objectives on the motivation example.

Data Associations	Conf.	Integ.	Avail.	N.-rep.	Auth.
Claim	0	0	1	1	0
Hospital details	0	1	1	0	1
Incident details	0	1	1	0	1
Hospital report	1	1	0	1	1
Police report	1	1	0	1	1
Medical report	1	1	0	1	1
Expert report	1	1	0	1	1
Account	1	1	0	0	1
Amount	1	1	0	1	1

As business process deployment is task-centric, these values need to be translated into security needs on tasks. We use the maximum values of the input and output objects of a task. For example, *Obtain Incident Details* is associated to the data objects *Claim*, *Incident Details* and *Hospital Details*. So, by taking the maximum of each data object's need we get the need of *Obtain Incident Details*: {0, 1, 1, 1, 1} for respectively {$Confidentiality$, $Integrity$, $Availability$, $Non\text{-}Repudiation$, $Authenticity$}.

Risk evaluation for each provider In this paper we take into account five cloud security threats given by the CSA [3] to evaluate the risk. These threats are each related to the 5 CIANA objectives:

- *Data Breaches* = {*Confidentiality*}
- *Data Loss* = {*Availability, Non-Repudiation*}
- *Account Hijacking* = {*Confidentiality, Integrity, Availability, Non-Repudiation, Authenticity*}
- *Insecure Interfaces* = {*Confidentiality, Integrity, Authenticity*}
- *Denial of Service* = {*Availability*}

Harm - We combine these relations with the security needs of each data object to obtain a so-called **harm**. The **harm** is defined on each data object, for each threat through the sum of the affected security needs. For example, the *Insecure Interfaces* threat (t) has the following harm on *Obtain Incident Details* (ta):
$Harm(t, ta) = (1 \times 0) + (1 \times 1) + (0 \times 1) + (0 \times 1) + (1 \times 1) = 3$

The first value of each bracket is equal to 1 if the threat is related to the objective, 0 otherwise, and the second value is the need calculated previously. Notice that the harm value is in this case always between 0 and 5. The result for all tasks of the process can be seen in Table 2.

Coverage - Now that we have the harm of a threat on each task of the process, we need to determine the response to these threats for each cloud provider. For this information we use the STAR Registry and the matrix defined by the CSA [3].

The CSA matrix defines a list of security controls a cloud provider should implement to reduce security risks. Each of these controls can be related to one or multiple threats. For example, the control "*IS-19.4 - Do you maintain key management procedures ?*" mitigates the *Data Breaches* threat. This matrix defines a total of 197 controls to implement in order to respond to all threats in the best possible way.

The STAR Registry publishes the list of implemented controls for providers willing to follow these recommendations. These lists are freely accessible and can help to check if a control has been put in place by a specific provider or not.

In our case, we use these two information as binary values (a control mitigates a threat or not / a control is implemented by a provider or not) to calculate a so-called **coverage score**, which indicates the response of a provider to a given threat. This value is a percentage, if the provider implements all controls mitigating a threat, it gets a coverage for this threat of 100%. In our case, this percentage is brought to a score on a scale of 0 to 5 (with 5 equivalent to 100%). We took 5 providers from the registry and calculated their coverage score, it is available in Table 2.

Table 2. Harms on the process tasks and coverage of the providers for 5 cloud threats

	Initiate claim	**Obt. incident det.**	**Obt hospital rep.**	**Obt. police rep.**	**Obt. expert rep.**	**Send reimb. dec.**	**Process reimb.**	**Softlayer**	**CloudSigma**	**FireHost**	**SHI Intern.**	**Terremark**
	Harm							Coverage				
Data breaches	1	0	1	1	1	1	1	2	5	2	4	4
Data loss	2	2	2	2	1	1	1	3	4	2	4	5
Account Hijacking	5	4	5	5	4	4	4	4	5	3	5	3
Insecure interfaces	3	2	3	3	3	3	3	4	5	3	4	4
Denial of service	1	1	1	1	0	0	0	4	5	4	5	5

Too risky provider exclusion Usually, the *vulnerability* is assessed and used to calculate a risk value of an information system. But in a cloud context, providers may be tempted to conceal their vulnerabilities for security reasons. This is why we use the **coverage** based on the security controls. By using the maximum possible coverage value $Covg_{max}$ (in our case 5), it is possible to get an equivalent to the vulnerabilities. Therefore, by combining this value with the **harm** we can define the following risk formula for a threat t, a task ta and a provider p:
$Risk(t, ta, p) = Harm(t, ta) + (Covg_{max} - Covg(p, t))$

Table 3 shows the maximum risk value for the five CSA threats, for the tasks of our motivating example, on five different providers (*Softlayer*[4], *CloudSigma*[5], *FireHost*[6], *SHI Int.*[7] and *Terremark*[8]).

Table 3. Maximum risk value of the tasks for each provider

	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark
Initiate claim	6	5	7	5	7
Obt. incident det.	5	4	6	4	6
Obt. hospital rep.	6	5	7	5	7
Obt. police rep.	6	5	7	5	7
Obt. expert rep.	5	4	6	4	6
Send reimb. dec.	5	4	6	4	6
Process reimb.	5	4	6	4	6

In accordance with the consumer, the broker defines the level of acceptable risk (referred to as **threshold**). For a given task, this threshold defines the providers with a too high risk value and excludes these deployment options. In our example, we set the threshold to 5, the cells of eliminated providers are grayed out in Table 3. Respectively a white cell means that the task can be deployed on the provider.

3.2 Cloud Selection

We select the target cloud environments in two stages: different configurations evaluation and final clouds selection.

Configurations evaluation We need to evaluate the different possible deployment configurations. To do this, we introduce a cost model which will allow us to balance the risks against the costs.

Cost model - Our cost model takes into account three types of costs:

- **Usage costs**, the CPU power needed to execute the process ($/GHz/month). The need is annotated on the tasks of the process.
- **Storage costs**, the space needed by the data of the process ($/GB/month). The size is annotated on the data objects of the process.
- **Transfer costs**, the amount of incoming/outgoing messages ($/GB). This size is calculated with the data exchanged between the process fragments.

Table 4 gives costs for the selected cloud offers of our motivating example.

[4] `http://www.softlayer.com`
[5] `http://www.cloudsigma.com`
[6] `http://www.firehost.com`
[7] `http://www.shi.com`
[8] `http://www.terremark.com`

Table 4. Costs of 5 Cloud offers

	Usage ($/GHz/mo)	Storage ($/GB/mo)	Transfer ($/GB)
Softlayer	20.00	0.10	0.10
CloudSigma AG	13.86	0.18	0.06
FireHost	25.70	2.78	0.50
SHI International, Corp.	11.56	0.29	0.01
Terremark	3.60	0.25	0.17

To find a deployment configuration (which task will be enacted on which cloud) we use an heuristic approach described in [6].

For our motivating example we tested our algorithm in three different ways. The results are shown in Table 5.

First run has no restrictions regarding the risk value, only the costs are taken into account. This gives us a "cheap" solution while leaving out security.

Second run includes a global risk threshold of 6. The majority of the tasks are now located on a more expensive but also less "risky" offer.

Third run has a global risk threshold set to 6. Once again, the solution becomes more expensive, but could be considered more "secure" than the second run.

We can notice that the **transfer costs** conduct to a regrouping of the tasks on one main offer to restrain the global costs.

Table 5. Output for different runs

	First run					Second run					Third run				
	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark
Initiate claim				x				x				x			
Obt. incident det.				x					x					x	
Obt. hospital rep.			x					x				x			
Obt. police rep.				x				x				x			
Obt. expert rep.				x				x				x			
Send reimb. dec.				x				x				x			
Process reimb.				x				x				x			
Risk	7					6					5				
Cost ($/mo)	84.58					206.68					259.24				

For those tests we optimized the costs while constraining the risk, but the algorithms can be extended to optimize the risk and constrain the cost, or even take into account other criteria.

Final configuration selection These resulting deployment configurations are analyzed by the cloud broker in conjunction with the cloud consumer to select the most adequate one. For our motivating example, we select the *Third run*.

3.3 Process deployment in the cloud

The process is deployed on the target environments in two steps: process decomposition and fragments deployment.

Process decomposition According to the selected configuration, the process is decomposed in multiple process fragments. A fragment is an autonomic business process enacted on one cloud and includes additional *synchronization tasks*. These additional tasks support the collaboration with the remote fragments to guarantee the control flow of the initial process. More details can be accessed in [7] and in [8].

The decomposed motivating example according to the selected configuration is depicted in Fig.3 (grey activities are *synchronization tasks*).

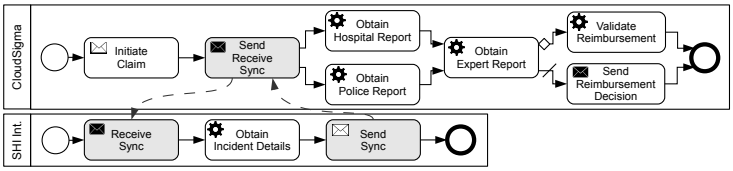

Fig. 3. A partitioned orchestration process.

Deployment in clouds The last step of our approach consists in deploying this configuration on the selected cloud offers. In [8] we presented how we deploy such service composition as BPEL programs on remote service orchestration engines (e.g., Apache ODE[9]).

4 Related Work

In [9] the authors present methods to distribute applications on different cloud environments. However security aspects are not considered. The authors of [2] are adapting processes through risk-reduction patterns, and in [13] processes are analyzed to decide if they are ready for a cloud deployment or not. But these two methods do not show the calculation of the risk value and do not consider process fragmentation.

In opposition to our proposal, [4] provides a risk-prediction algorithm to help users to take decisions during the execution of the process. We focus on design-time rather than on runtime, which changes slightly the kind of treated risks.

The model presented in [11] allows to evaluate security vulnerabilities in a Service Oriented Architecture, but does not take into account the cloud context. Our coverage approach based on security controls given in different standards ([1], [3]) seems to be more adapted to such a context.

Watson [12] decomposes workflows and deploys them on multiple clouds according to a cost model, but he defines arbitrary security levels for each provider. A more complete cost model is presented in [10], but it is not easily adaptable in the business process context for an automated treatment as it is done in our approach.

[9] `http://ode.apache.org/`

5 Conclusion and Future Work

In this paper we have presented a cloud broker framework for assessing security risks in a multiple-cloud context. We assess security risks of cloud providers using standard-based and industry accepted security controls and risk listings. We focus on one business process to illustrate how these risk values, in combination with costs, can help a cloud broker to take decisions for the cloud provider selection. The paper demonstrates the feasibility of our approach with a motivating example and real cloud providers.

Some limitations are not addressed in this paper. First, the shortage of empirical evaluation on real use cases, which will be realized in future works with domain experts and industrial partners. Another point is that our approach takes place at design-time, but as the Cloud is a very dynamic context, extending our framework to configuration at run-time would be an interesting improvement. Finally, the binary values for the *security needs*, *mitigations* and *control implementation* could be replaced with more complete scales, as some security controls "better" mitigate threats than others.

References

1. ISO/IEC 27017, Information tech., Security techniques, Code of practice for information security controls for cloud computing services based on ISO/IEC 27002.
2. O. Altuhhova, R. Matulevicius, and N. Ahmed. Towards definition of secure business processes. In *CAiSE Workshops*, pages 1–15, 2012.
3. Cloud Security Alliance. Cloud Control Matrix / Security, Trust & Assurance Registry / Consensus Assessments Initiative Questionnaire. Technical report.
4. R. Conforti, M. de Leoni, M. L. Rosa, and W. M. van der Aalst. Supporting risk-informed decisions during business process execution. In *CAiSE'13*, pages 116–132, Valencia, Spain, 2013.
5. European Network and Information Security Agency. Benefits, risks and recommendations for information security. Technical report, 2009.
6. W. Fdhila, M. Dumas, and C. Godart. Optimized decentralization of composite web services. In *CollaborateCom'10*, pages 1–10, 2010.
7. W. Fdhila, U. Yildiz, and C. Godart. A flexible approach for automatic process decentralization using dependency tables. In *ICWS '09*, pages 847–855, Washington, DC, USA, 09. IEEE Computer Society.
8. E. Goettelmann, W. Fdhila, and C. Godart. Partitioning and cloud deployment of composite web services under security constraints. In *IC2E'13*, 2013.
9. F. Leymann, C. Fehling, R. Mietzner, A. Nowak, and S. Dustdar. Moving applications to the cloud: an approach based on application model enrichment. *IJCIS*, 20(3):307–356, 2011.
10. B. Martens, M. Walterbusch, and F. Teuteberg. Costing of cloud computing services: A total cost of ownership approach. In *ICSS'12*, pages 1563–1572, 2012.
11. S. Sackmann, L. Lowis, and K. Kittel. A risk based approach for selecting services in business process execution. In *Wirtschaftsinformatik (1)*, pages 357–366, 2009.
12. P. Watson. A multi-level security model for partitioning workflows over federated clouds. In *CloudCom*, pages 180–188, 2011.
13. S. Wenzel, C. Wessel, T. Humberg, and J. Jürjens. Securing processes for outsourcing into the cloud. In *CLOSER*, pages 675–680, 2012.