A Formal 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
claude.godart@loria.fr
[2] CRP Henri Tudor, Kirchberg, Luxembourg, Luxembourg
{elio.goettelmann,benjamin.gateau}@tudor.lu
[3] Blu Age - Netfective Technology, Pessac, France
k.dahman@bluage.com

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 services, at different prices, and with various quality levels. Cloud computing can reduce costs, but the selection of a solution adapted to one's needs is time consuming. For this purpose, cloud brokers have emerged; they propose to help cloud consumers for selecting adequate solutions. They compare existing offers, essentially against their prices. In this selection process, security still is an important factor. Since cloud computing presents new kinds of security risks [6,9], they need to be tamed before a wider adoption. Novel methods have to be defined in order to prevent potential losses on companies.

Distributing software over multiple locations increases the complexity of gathering sensitive business information. In this paper, we propose a framework for cloud brokers. We propose to help analyzing the security levels of different cloud offers by following standard risk assessment methodologies.

The paper is organized as follows. Section 2 presents a motivating example used to demonstrate the purpose and the scope of our framework. In Sect. 3

S. Nurcan and E. Pimenidis (Eds.): CAiSE Forum 2014, LNBIP 204, pp. 3–19, 2015.
DOI: 10.1007/978-3-319-19270-3_1

we present our model for assessing security risks in a cloud context. Section 5 applies our approach on the motivating example and gives our experimental results. Section 6 gives a brief description of our proof-of-concept implementation. Sections 7 and 8 discuss respectively related and future work.

2 Motivating Example and Overview

In this section, we introduce a motivating example to illustrate the concepts of our framework. We give an overview of our approach for selecting offers when deploying business processes on multiple clouds.

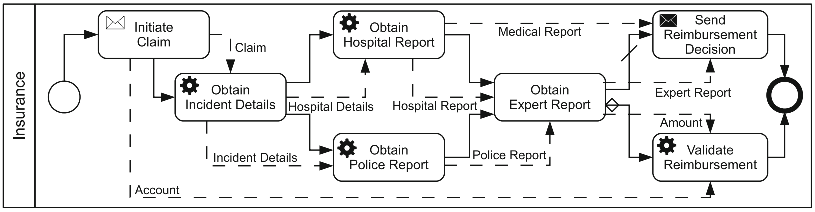

Fig. 1. Business process motivation example: Insurance Claim Recovery Chain.

Consider the *insurance claim recovery chain* [11] depicted in Fig. 1. This BPMN process is initiated when an *insurance* company receives a claim recovery declaration from a *beneficiary*. The emergency service is invoked to obtain details about the *incident*. The *hospital* and *police* reports are required by the *expert* to decide on the *reimbursement* decision. The *bank* is potentially requested and a notification is sent to the *beneficiary*.

In the figure, *data dependencies* between the different tasks are represented using dashed arrows. Additional requirements regarding the distribution of the process can be modeled. Examples of such requirements are described more precisely in [13].

Now suppose that this insurance company wants to outsource this process to the cloud. 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 approach consists in a design-time tool and a model-driven approach for producing secure and cost-effective business processes on multiple clouds. It is illustrated in Fig. 2.

A cloud broker requests the *security needs* as non-functional requirements from the cloud consumer. It analyzes different *cloud offers* of cloud providers. In this paper we focus on the **risk assessment**. Our goal is to show how the risk can be assessed in a cloud context.

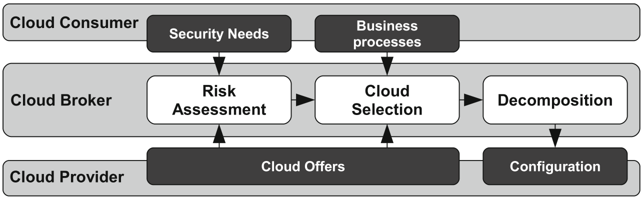

Fig. 2. Our design-time framework for multi-cloud business process deployment.

The broker uses *business processes* of the cloud consumer to **select** the adequate (secure and cost-effective) *cloud offers* based on the functional requirements, the costs and the previously calculated risk.

The broker **decomposes** the business process into smaller parts, as each task can be enacted on a different cloud site. The generated *configuration* is the assignment of these process fragments to cloud provider sites. The decomposition itself has already been addressed in [12].

In the next section we will define the different concepts of our approach.

3 A Formal Cloud Security Risk Assessment Model

In the following, we present our model (Fig. 3) for assessing security risks in a cloud context. We show examples of a formalization of each concept and their relations.

In traditional risk management methodologies, the risk is generally evaluated using the following formula: $risk = vulnerability \times impact \times threat$ [2,17]. The *threat* represents the event which would negatively affect the information system. The *impact* represents the loss (financial or other) which would occur for this event. The *vulnerabilities* represent the security flaws which could enable this event.

But in a cloud environment, the information system owner is separated from its user. In this context it is difficult to assess the risk in this way. Indeed, the cloud provider cannot determine the impact of a security breach on its consumer's system. The cloud consumer cannot identify easily the vulnerabilities of the cloud provider's infrastructure. Moreover, cloud providers often conceal their vulnerabilities to reduce their exposure to attacks.

Therefore, the cloud broker plays a crucial role. It is the sole stakeholder who can calculate a risk value by taking into account information from providers and consumers. Our model is divided into three packages:

- **Cloud Consumer Model**, for establishing the *impact* value of the risk.
- **Cloud Provider Model**, for establishing the *vulnerability* value.
- **Cloud Broker Model**, which combines these concepts with the *threats* to evaluate a final *risk* value.

In the following we furtherly investigate the model of each stakeholder.

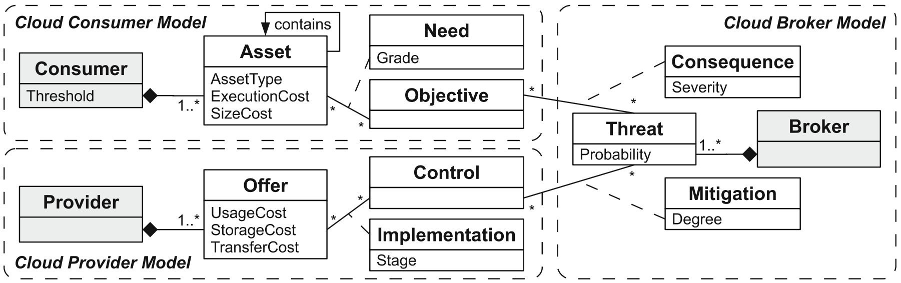

Fig. 3. Cloud security risk assessment model

Definition 1 (Cloud Consumer). *An entity which browses the offers from Cloud providers (or Cloud brokers) and uses the adequate service for its functional and non-functional requirements. It is formalized as follows:*

Asset = *Process* ∪ *Task* ∪ *Data* ∪ *Message, defines business assets of the consumer information system,*
Objective = {*Confidentiality, Integrity, Availability, Non-repudiation, Authenticity*}*, is a set of security objectives on the information system,*
Need *: Asset* × *Objective* → *Grade, defines the security need of an asset and an objective,*
Threshold *: Ω* → *Level, defines a global acceptable risk level,*
ExecutionCost : Task → $\mathbb{N}^*$*, defines task execution costs (in seconds),*
SizeCost : Data → $\mathbb{N}^*$*, defines data size costs (in bytes),*

A **Security Need** is defined as a **Grade** on an **Asset** for a **Security Objective**. For our approach we use the widespread CIANA security objectives (*Confidentiality*, *Integrity*, *Availability*, *Non-Repudiation* and *Authenticity*). For example, it is more important for some assets to be often available, whereas for other priority is more about confidentiality. Such security requirements can be described with these attributes. The grade defines "how much" of a security objective the asset needs. It can be adjusted regarding the use case but often corresponds to a qualitative or quantitative scale (see Table 1). Other reference frames can also be used as security objectives, for example STRIDE[1].
These needs can be annotated on the data elements of a business process and are then translated into task-centric needs, or they can be directly annotated on the tasks.

The **Threshold**, indicates the **Level** of acceptable risk for the information system.

Definition 2 (Cloud Provider). *An entity which can hold multiple cloud offers. It is responsible for the implementation of security controls, which protects its offers from attacks and complies with regulations. It is formalized as follows:*

[1] http://msdn.microsoft.com/en-us/magazine/cc163519.aspx.

Offer, *is a set of cloud offers,*
Control, *is a set of controls,*
Implementation*: Offer* × *Control* → *Stage, defines how a control is implemented for an offer,*
UsageCost, StorageCost, TransferCost : *Offer* → $\mathbb{N}^*$*, defines offers' costs in terms of usage (in \$/second), storage (in \$/byte) and transfer (in \$/byte)*

Security Controls are countermeasure reducing vulnerabilities of information systems. They may be given in guidelines or standards (as for example the CSA [6] or the ISO 27017 [1]). Providers can then audit the **implementation** of these controls for their **Offers**. Since some control implementations can be described by a fuzzy value, we add a **Stage** attribute. It expresses the fact that some controls are not fully implemented but only partially, which is still better than not at all. We consider that a cloud provider is more willing to publish such type of information than its vulnerabilities. It gives information about the security of their information system without directly revealing their vulnerabilities (e.g. the CSA Security, Trust and Assurance Registry: STAR [6]).

Definition 3 (Cloud Broker). *An entity that enhances existing services like security, combines multiple services or measures different providers and selects the best [18]. It is formalized as follows:*

Threat, *is a set of security threats, which can be weighted through a Probability,*
Control, *is the same set of controls as the Cloud provider,*
Objective, *is the same set of objectives as the Cloud consumer,*
Mitigation*: Threat* × *Control* → *Degree, is the mitigation of a threat by a control,*
Consequence*: Threat* × *Objective* → *Severity, is the consequence of a threat on an objective,*
Harm*: Threat* × *Asset* → *Rate, is the harm a threat could have on an asset,*
Coverage*: Threat* × *Offer* → *Score, is the coverage score of a threat for an offer,*
Risk*: Asset* × *Offer* × *Threat* → *Level, is the risk level of a threat for an asset on an offer,*

To stick to the previous risk definition, the broker defines a list of **Threats** the Cloud Consumer faces. The CSA regularly publishes an example of such a list [6]. As some threats are more likely to happen, we added a **Probability** of occurence. This ponderation depends on the context and on the type of cloud offer. For example, some companies are more exposed to hacking attacks than others or some threats may not be applicable for some cloud offers.

The **Mitigations** represent relations between controls and threats. Each control reduces one or more threats. But as some controls may have a bigger reduction effect, we introduce a **Degree** of mitigation. This value indicates how the control mitigates the given threat.

By combining this value with the **Implementation**, we calculate a **Score** that defines the threat **Coverage** for a provider. Basically, the more controls a provider implements, the more threat he will cover. But as some controls are "better" to mitigate some threats, the provider will have a different response for each threat. This is the value which allows the broker to differentiate the cloud providers.

Consequences represent the relation between objectives and threats. Each threat affects one or more security objectives. But as some objectives are more affected than others, we introduce the **Severity** of the consequence. It indicates how the objective is affected by the given threat. For example, a *Denial of Service* attack will affect the system's *Availability*, more than its *Confidentiality.*

We combine this value with the **Need** given by the consumer, to get the **Rate** of **Harm**. It represents the *impact* of previously given risk formula. For example, an asset that has an important need of *Availability* will have an important Harm in case of a *Denial of Service* attack.

Finally, we define the **Level** of **Risk** for an asset, an offer and a given threat. It can be calculated by the broker through a combination between the Harm, the Coverage and the Probability. In the next section we give an example of such a risk level.

4 Instantiation on the Motivating Example

This section instantiates our model on the motivating example. For this purpose, we specify the relations between the different concepts defined previously. To simplify relations and calculations we define all attributes on a $[0, 1]$ scale and work with a probabilistic approach.

4.1 Calculating the Harm

First, we calculate the Harm, which represents the *impact* in the commonly used risk formula. It is defined between the Cloud Consumer and the Cloud Broker.

Security Needs. As presented previously, the security needs are described using the five common CIANA attributes.

Table 1. Grades of need for each security objective

Confidentiality	Integrity	Availability	Non-repudiation	Authenticity	**Values**
Public	Passable	Sparse	Futile	Irrelevant	**0**
Restricted	Alterable	Usual	Tolerable	Common	**0.5**
Secret	Fixed	Continuous	Trusted	Verified	**1**

We define 3 grades for each objective and assign them a value in Table 1. A value of 0 means the objective is not needed at all and 1 that this objective is crucial.

Table 2. Annotations of the Security Needs on the motivation example's data.

Data Associations	Conf.	Integ.	Avail.	N.-rep.	Auth.
Claim	Public	Passable	Continuous	Tolerable	Irrelevant
Hospital details	Secret	Alterable	Usual	Futile	Common
Incident details	Public	Alterable	Usual	Futile	Common
Hospital report	Restricted	Fixed	Sparse	Tolerable	Common
Police report	Restricted	Fixed	Usual	Trusted	Verified
Medical report	Secret	Alterable	Usual	Tolerable	Verified
Expert report	Restricted	Fixed	Usual	Trusted	Verified
Account	Secret	Fixed	Sparse	Futile	Verified
Amount	Restricted	Fixed	Sparse	Trusted	Verified

For our motivating example, Table 2 shows the *security needs* on each data object. These security needs are negotiated by a risk manager with the cloud consumer. For example, the payment does not need a high *Availability* since it is sensible but not urgent. It will be not as much exposed to a *Denial of Service Attack* than another activity.

Business process deployment is task-centric, as tasks are assigned to cloud offers and not data objects. Therefore, 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, similar to the approach presented by Watson in [22]. For example, *Obtain Incident Details* is associated to the data objects *Claim*, *Incident Details* and *Hospital Details*. We take the maximum of each data object's need to get the need of *Obtain Incident Details*: $\{Secret, Passable, Continuous, Tolerable, Common\}$ for respectively $\{Confidentiality, Integrity, Availability, Non\text{-}Repudiation, Authenticity\}$. The security needs of all tasks are given in Table 3.

Consequences. We advocate a value for the **Severity** between 0 and 1. The minimum 0 means that the objective is not affected at all. The maximum 1 means that the objective is completely prevented by the given threat.

Table 3. Resulting Security Needs on the motivating example's tasks

Tasks	Conf.	Integ.	Avail.	N.-rep.	Auth.
Initiate Claim	Secret	Fixed	Continuous	Tolerable	Verified
Obtain Incident Details	Secret	Alterable	Continuous	Tolerable	Common
Obtain Hospital Report	Secret	Fixed	Usual	Tolerable	Verified
Obtain Police Report	Restricted	Fixed	Usual	Trusted	Verified
Obtain Expert Report	Restricted	Fixed	Usual	Trusted	Verified
Send Reimbursement Decision	Secret	Fixed	Usual	Trusted	Verified
Process Reimbursement	Secret	Fixed	Sparse	Trusted	Verified

Table 4. Severity of consequences for each security objective

	Conf.	Integ.	Avail.	Non-Rep.	Auth.
Data breaches	Significant	Negligible	Negligible	Negligible	Negligible
Data loss	Negligible	Negligible	Significant	Related	Negligible
Account hijacking	Significant	Significant	Related	Related	Related
Insecure interfaces	Significant	Significant	Related	Negligible	Significant
Denial of service	Negligible	Negligible	Significant	Negligible	Negligible
Malicious insiders	Significant	Significant	Significant	Significant	Significant
Abuse of Cloud Services	Negligible	Related	Negligible	Related	Related
Insufficient Due Diligence	Related	Related	Significant	Negligible	Negligible
Technology Vulnerabilities	Related	Related	Related	Related	Related

For our motivating example we define three levels ($\{Significant = 1, Related = 0.5, Negligible = 0\}$) and assign them in Table 4. These threats and relations are based on the CSA report [7].

Harm. To calculate the harm we use a probabilistic approach to remain in the $[0, 1]$ interval. We want to express that the harm is the union of the effects on all objectives. We define the formula:

Definition 4 (Harm). $\forall t \in Threat, \forall a \in Asset,$

$$Harm(t, a) = 1 - \left(\prod_{o_i \in Objective} 1 - (Need(a, o_i) \times Consequence(t, o_i)) \right) \quad (1)$$

Our framework also works with other types of combination strategies as weighted or mixed, which could be more adapted in other use cases. The result for all tasks of the process can be seen in Table 5.

4.2 Calculating the Coverage

Then, we calculate the Coverage, which represents the *vulnerability* in the commonly used risk formula. It is defined between the Cloud Provider and the Cloud Broker.

Implementation. In order to determine the response to cloud threats for each offer we use Security, Trust and Assurance Registry (STAR) [6] managaed by the CSA. The site publishes a list of major public cloud providers and the controls they implement. We define the **Stage** of **Implementation** over three levels: $\{Full = 1, Partial = 0.5, Ignored = 0\}$. As there are 197 implementable controls, we do not publish an exhaustive list for each provider in this paper.

Table 5. 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	Probability
	Harm							Coverage					
Data breaches	1.00	1.00	1.00	0.50	0.50	1.00	1.00	0.36	0.62	0.21	0.54	0.52	1.0
Data loss	1.00	1.00	0.63	0.75	0.75	0.75	0.50	0.49	0.59	0.42	0.59	0.67	1.0
Account Hijacking	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.55	0.64	0.46	0.59	0.46	1.0
Insecure interfaces	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.54	0.67	0.53	0.59	0.58	0.9
Denial of service	1.00	1.00	0.50	0.50	0.50	0.50	0.00	0.60	0.66	0.53	0.62	0.66	0.9
Malicious insiders	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.49	0.64	0.54	0.60	0.61	0.9
Abuse of Cloud Services	0.81	0.58	0.81	0.88	0.88	0.88	0.88	0.58	0.60	0.56	0.54	0.59	0.8
Insufficient Due Diligence	1.00	1.00	0.88	0.81	0.81	0.88	0.75	0.53	0.64	0.51	0.60	0.56	0.8
Technology Vulnerabilities	0.95	0.89	0.93	0.93	0.93	0.95	0.94	0.49	0.64	0.50	0.55	0.54	0.8

Mitigations. The CSA matrix recommends a list of security controls that 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. We suggest a value for the **Degree** in the interval $[0, 1]$. The minimum 0 means that the control is completely ineffective for mitigating the given threat, the maximum 1 means that the control completely prevents the given threat. Optimally, the sum of all weightings for a given threat, should be equal to 1 (when following strictly the CSA guidelines the coverage should be maximal). Moreover, the broker can decide to assign the weightings differently. Indeed, some threats may not be completely coverable. There would always be a remaining risk residue even when implementing all controls. Or he can decide that some controls become useless when implementing other ones. As it would not be really relevant to our proposal and due to a lack of space, this mitigation matrix is not shown in this paper.

Coverage. We take a probabilistic approach to calculate the Coverage Score. We want to express that the coverage is the union of the effects of all controls. Thus, we remain in the same $[0, 1]$ interval. We defined the following formula:

Definition 5 (Coverage). $\forall t \in Threat, \forall o \in Offer$

$$Coverage(t, o) = 1 - \left(\prod_{c_i \in Control} 1 - (Implementation(o, c_i) \times Mitigation(t, c_i)) \right) \tag{2}$$

Note that, other types of combination strategies can be used without impacting our previously defined model. For example, in some use cases a mixed or a weighted strategy could be more adapted.

The results with this formula on 5 providers selected from the STA Registry are shown in Table 5 (*Softlayer*[2], *CloudSigma*[3], *FireHost*[4], *SHI Int.*[5] and *Terremark*[6]). It also indicates the probability we use for each threat. It follows the ranking given by the CSA [7].

4.3 Calculating the Security Risk

To assess the final risk value, **Harm**, **Coverage** and the Threat **Probability** are combined. We use the complementary of the coverage to comply to the commonly used risk formula (given in Sect. 3) which uses the *vulnerabilities*.

Definition 6 (Risk). $\forall a \in Asset, \forall o \in Offer, \forall t \in Threat,$

$$Risk(a, o, t) = Harm(t, a) \times (1 - Coverage(t, o)) \times Probability(t) \qquad (3)$$

Table 6. Maximum risk value of the tasks for each provider

	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark
Initiate claim	0.64	0.41	0.79	0.46	0.54
Obt. incident det.	0.64	0.41	0.79	0.46	0.54
Obt. hospital rep.	0.64	0.38	0.79	0.46	0.54
Obt. police rep.	0.46	0.36	0.54	0.41	0.54
Obt. expert rep.	0.46	0.36	0.54	0.41	0.54
Send reimb. dec.	0.64	0.38	0.79	0.46	0.54
Process reimb.	0.64	0.38	0.79	0.46	0.54

With this formula, our security risk value remains in a $[0, 1]$ interval. It can then be brought to levels if needed, for example: $\{0 \leq Low \leq 0.3 < Medium < 0.7 \leq High \leq 1\}$. We select for each asset on each offer the highest risk value and present them in Table 6.

In accordance with the consumer, the broker defines the **Threshold**, the level of acceptable risk. For a given task, this value defines the providers with a too high risk and excludes these deployment options. In our example, we set the threshold to 0.5. In Table 6 the cells of eliminated providers are grayed out. Respectively a white cell means that the task can be deployed on the provider.

Note that two providers (FireHost and Terremark) are completely excluded, as they do not sufficiently cover threats in comparison to the other providers. Two other providers (CloudSigma and SHI Int.) can be used for enacting all tasks of the example process. They implement enough controls for presenting an acceptable security risk. The last provider (Softlayer) can only be used for

[2] http://www.softlayer.com.

[3] http://www.cloudsigma.com.

[4] http://www.firehost.com.

[5] http://www.shi.com.

[6] http://www.terremark.com.

deploying two tasks, as the others have to high security requirements for this provider.

The next section shows how the final configuration is chosen among the remaining offers.

5 Experimentation and Evaluation

In this section we go back to our global approach to deploy the example process in a secure and cost-effective way on multiple clouds. First we select the final configuration based on the costs, and then we deploy the process on the chosen offers.

5.1 Cloud Selection

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

Configurations Evaluation. To evaluate the different possible deployment configurations, we introduce a cost model. It allows us to balance the risks against the costs.

Cost model - We consider 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.

When benchmarking different existing cloud providers we noticed that their pricing schemes generally match these types of costs. More complex pricing plans can often be brought to such a cost distinction.

Table 7 gives costs for the selected cloud offers of our motivating example.

Table 7. 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

The motivating example presented in this paper is a quite simple case study, as there are only seven tasks to assign to a pool of five cloud providers. But in other use cases, this assignment problem can rapidly explode leading to a Quadratic Assignment Problem (QAP) (n tasks to p providers). Thus, to find a good solution in an acceptable time, we use an heuristic approach. It is not described in details in this paper, as it is not the goal of our proposal, but details can de found in [10]. Basically, it consists in finding an initial solution (a so-called *Greedy* solution) that we enhance using a *Tabu* search algorithm.

Table 8. Output for different runs

	First run					**Second run**					**Third run**					**Fourth run**				
	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark
Initiate claim					x				x			x					x			
Obt. incident det.					x				x			x					x			
Obt. hospital rep.					x				x			x					x			
Obt. police rep.					x				x					x			x			
Obt. expert rep.					x				x					x			x			
Send reimb. dec.					x				x			x					x			
Process reimb.					x				x			x					x			
Risk	0.54					0.46					0.41					0.41				
Cost ($/mo)	67.65					203.38					232.28					275.13				

We experimented our algorithm on the motivating example in four different ways. The results are shown in Table 8.

First run has no restrictions regarding the security risk value, only the costs are taken into account. This gives us a “cheap” solution while leaving out security. In this run, the risk calculated in the previous section is completely disregarded and no providers are excluded. When compared to other possible solutions it can give an idea about the “costs of security”.

Second run includes a global risk threshold of 0.5. The providers excluded in the previous section cannot be selected. 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 0.45. By decreasing the acceptable risk value, the provider of the previous run can no longer be an option for some tasks. These tasks are moved to a more “secure” offer and thus increasing the global “security level”. Obviously, the costs of the configuration are also increasing.

Fourth run presents a different approach, where we all tasks are moved to the “most secure” cloud. As this provider seems to be the best in terms of security (it has the lowest security risk values), it gives an idea about the "best" possible configuration when disregarding costs. But the risk value indicates that there is no real improvement in comparison to the previous run. Only the costs of the configuration increases, which makes this possible solution not really interesting.

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

Another point is that it is possible that no configuration is found if the threshold risk value is too low. In this case, either the user increases the threshold value, or he considers the Cloud context as too risky and decides not to move the process to the Cloud.

Final Configuration Selection. The cloud broker can analyze the deployment configurations our algorithm produces and select the most adequate one in conjunction with the cloud consumer. For our motivating example, we select the *Third run*. Indeed, in comparison to the **Second run** it brings a non-negligible improvement in terms of security and the increase of costs seems to us as acceptable. In the following we present briefly how the process can then be deployed on these two selected cloud offers.

5.2 Process Deployment in the Cloud

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

Process Decomposition. We decompose the process in two process fragments according to the selected configuration. A fragment is a business process enacted on one cloud and includes additional *synchronization tasks*. These tasks support the collaboration with the remote fragments to guarantee the control flow of the initial process. Please refer to [11,12] to see more details about that.

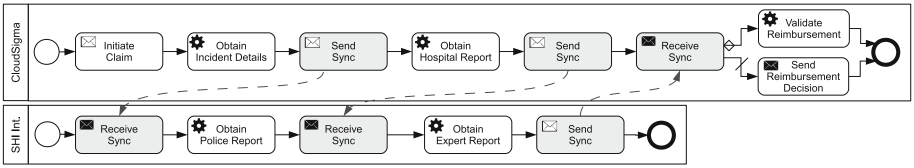

Fig. 4. The fragmented example process.

The decomposed motivating example according to the selected configuration is depicted in Fig. 4 (grey activities are *synchronization tasks*). The two tasks *Obtain Police Report* and *Obtain Expert Report* form an autonomous process located on a different cloud than the remaining process. This process fragment can execute when it receives the *synchronisation* message sent from the first fragment. These two fragments form a global process equivalent to the original centralized process. This global process ends when both fragments have reached their ending tasks.

Deployment in Clouds. The last step of our approach consists in deploying these fragments on the selected cloud offers. In [12] we presented how we deploy such service composition as BPEL programs on remote service orchestration engines (e.g., Apache ODE[7]). But as current Cloud offers do not support this kind of services, we selected offers providing infrastructure services and used them to deploy our own execution engines. We are confident that such type of platforms will rapidly emerge in the future, WSO2 Stratoslive[8] is such a PaaS even if still not available through web services. For some type of processes, tasks could be mapped to existing offers at the SaaS layer of the Cloud. For others, not service based, a development phase can be required before deploying them on the Cloud.

6 Proof of Concept Implementation

To show the feasibility of our approach we implemented our model in a web tool. It consists in a PHP application using the Laravel[9] framework and running on a web server with a MySQL database.

The user of the tool can add new assets, associate security objectives to them and assign the according **security needs** through a dropdown menu. Based on the given values the risk level for each provider is calculated and presented on a chart (see Fig. 5). The risk values a grouped by the type of threat and each provider is assigned to a specific color. The interface can also be adapted to group the values by providers and assign different colors to the different threats. In this case it becomes easy to see which providers are compliant to the given threshold. On this interface the user can also define a threshold which will filter out the offers with a too high risk value (the horizontal dotted line on Fig. 5).

Currently the risk is calculated through static database views, so, in order to add a new type of risk calculation or new levels for defining the security needs, a new view must be created manually. Even if it is sufficient for a prototype implementation and adapted for a demonstration purpose, we plan to extend our tool to support full customization from the user. Therefore, the user of the tool will be able to completely define how the risk will be calculated and re-adapt it "on-the-fly" to make it correspond to its needs.

7 Related Work

Fragmentation and multi-cloud deployment for increasing security is a hot topic in current research. Jensen [14] was one of the first to propose the decomposition of applications and their distribution on different cloud environments. AlZain confirmed later this trend in [4] and highlighted the importance of multi-cloud environments for increasing security.

[7] http://ode.apache.org/.

[8] WSO2 Business Process Server, http://wso2.com/products/business-process-server.

[9] http://laravel.com/docs/4.0.

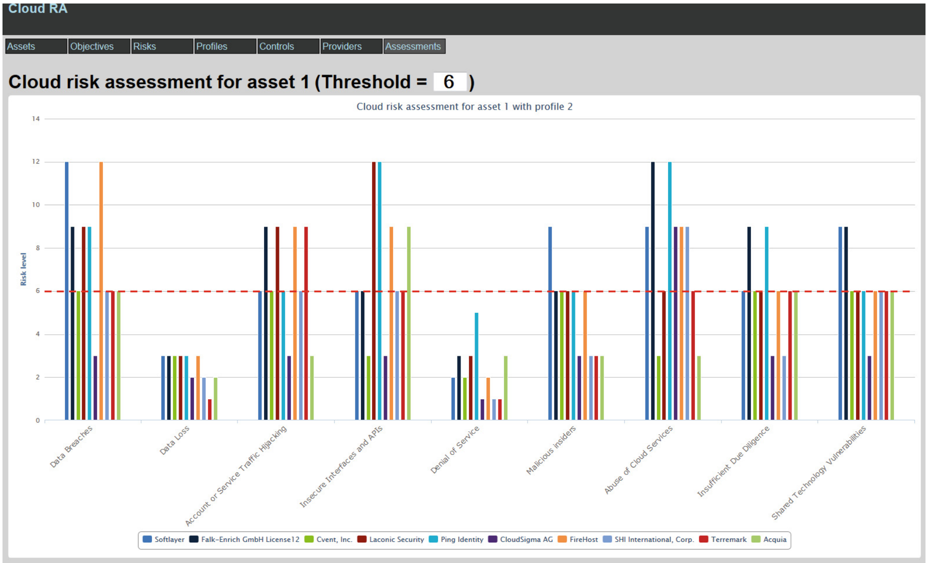

Fig. 5. Our proof-of-concept presenting the risk values for different cloud offers

In [15] the authors present methods to distribute applications on different cloud environments. The fragmentation can be done manually or use optimization algorithms. Similar to our approach the decomposition is considered in the early stages of business process modeling. However security aspects are not considered.

The authors of [3] are adapting processes through risk-reduction patterns, and in [23] 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.

Watson [22] decomposes workflows and deploys them on multiple clouds according to a cost model, but he defines arbitrary security levels for each provider. In our approach we take calculate these values with information obtained from real cloud providers. Another selection approach is presented in [5], where security requirements are matched with service capabilities. But it does not consider the business process globally and has not been designed for a cloud environment.

Otherwise, [8] provides a risk-prediction algorithm taking decisions during the execution of the process. We focus on design-time rather than on runtime, which changes slightly the kind of treated risks. This type of approach can be considered as complementary to our proposal.

The model presented in [20] allows to evaluate security vulnerabilities in a Service Oriented Architecture. Unfortunately, it does not take into account the cloud context. Our coverage approach is based on security controls from different standards specifically designed for the cloud [1,6]. It seems to us more adapted for such a context.

A more complete cost model is presented in [16]. It is not easily adaptable on business processes for an automated treatment, but could be used to refine our approach.

8 Conclusion and Future Work

In this paper we present a cloud broker framework for assessing security risks in a multiple-cloud deployment context. We assess security risks using standard-based and industry accepted security controls and risk listings. We focus on one use case to illustrate how these risk values, in combination with costs, can help cloud brokers to take decisions for the cloud provider selection. Our approach provides cost-effective and secure cloud deployment solutions. The paper demonstrates the feasibility of our approach with a motivating example and real cloud providers.

We define security needs with the five CIANA objectives and use the STAR Registry to calculate the threat coverage of cloud providers. However, as our approach is model driven, our tool can be extended to support other types of objectives, controls and threats.

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. We also argue that our security needs annotations on the business process could be furtherly explored. There are existing notations as BPMN extensions [19] or [21] that we could adapt to more precisely express the consumer's security requirements.

We are focusing our current work on the selection algorithm. Indeed, in this paper we select providers according to a single criterion, costs, while transforming the risk value in a constraint. However, multi-criteria optimization strategies could be more adapted, especially when taking into account more than two criteria. Quality of Service or response-time are also important parameters that need to be considered when outsourcing applications to the Cloud.

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. AS/NZS 4360 SET Risk Management, Australian/New Zealand Standards (2004)
3. Altuhhova, O., Matulevičius, R., Ahmed, N.: Towards definition of secure business processes. In: Bajec, M., Eder, J. (eds.) CAiSE 2012. LNBIP, vol. 112, pp. 1–15. Springer, Heidelberg (2012)
4. AlZain, M., Pardede, E., Soh, B., Thom, J.: Cloud computing security: from single to multi-clouds. In: HICSS 2012, pp. 5490–5499 (2012)

5. Carminati, B., Ferrari, E., Hung, P.C.K.: Security conscious web service composition. In: ICWS (2006)
6. Cloud Security Alliance. Cloud Control Matrix/Security, Trust & Assurance Registry/Consensus Assessments Initiative Questionnaire. Technical report
7. Cloud Security Alliance. The Notorious Nine - Cloud Computing Top Threats in 2013. Technical report (2013)
8. Conforti, R., de Leoni, M., La Rosa, M., van der Aalst, W.M.P.: Supporting risk-informed decisions during business process execution. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) CAiSE 2013. LNCS, vol. 7908, pp. 116–132. Springer, Heidelberg (2013)
9. European Network and Information Security Agency. Benefits, risks and recommendations for information security. Technical report (2009)
10. Fdhila, W., Dumas, M., Godart, C.: Optimized decentralization of composite web services. In: CollaborateCom 2010, pp. 1–10 (2010)
11. Fdhila, W., Yildiz, U., Godart, C.: A flexible approach for automatic process decentralization using dependency tables. In: ICWS 2009, pp. 847–855. IEEE Computer Society, Washington, DC (2009)
12. Goettelmann, E., Fdhila, W., Godart, C.: Partitioning and cloud deployment of composite web services under security constraints. In: IC2E 2013 (2013)
13. Goettelmann, E., Mayer, N., Godart, C.: A general approach for a trusted deployment of a business process in clouds. In: MEDES 2013 (2013)
14. Jensen, M., Schwenk, J., Bohli, J., Gruschka, N., Iacono, L.: Security prospects through cloud computing by adopting multiple clouds. In: CLOUD 2011, pp. 565–572 (2011)
15. Leymann, F., Fehling, C., Mietzner, R., Nowak, A., Dustdar, S.: Moving applications to the cloud: an approach based on application model enrichment. IJCIS **20**(3), 307–356 (2011)
16. Martens, B., Walterbusch, M., Teuteberg, F.: Costing of cloud computing services: a total cost of ownership approach. In: ICSS 2012, pp. 1563–1572 (2012)
17. National Institute of Standards and Technology. Information Security - Guide for Conducting Risk Assessments (2002)
18. National Institute of Standards and Technology. Cloud Computing Reference Architecture (2011)
19. Rodríguez, A., Caro, A., Cappiello, C., Caballero, I.: A BPMN extension for including data quality requirements in business process modeling. In: Mendling, J., Weidlich, M. (eds.) BPMN 2012. LNBIP, vol. 125, pp. 116–125. Springer, Heidelberg (2012)
20. Sackmann, S., Lowis, L., Kittel, K.: A risk based approach for selecting services in business process execution. Wirtschaftsinformatik **1**, 357–366 (2009)
21. Turki, S.H., Bellaaj, F., Charfi, A., Bouaziz, R.: Modeling security requirements in service based business processes. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Wrycza, S. (eds.) EMMSAD 2012 and BPMDS 2012. LNBIP, vol. 113, pp. 76–90. Springer, Heidelberg (2012)
22. Watson, P.: A multi-level security model for partitioning workflows over federated clouds. In: CloudCom, pp. 180–188 (2011)
23. Wenzel, S., Wessel, C., Humberg, T., Jürjens, J.: Securing processes for outsourcing into the cloud. In: CLOSER, pp. 675–680 (2012)