Implementing SLA Constraints in Multi-Cloud Environments

George-Valentin Iordache*, Bogdan Țigănoaia*, Cătălin Negru*
Florin Pop**

**Computer Science Department, University Politehnica of Bucharest, Bucharest, Romania*
(e-mail: george.iordache@cs.pub.ro, bogdantiganoaia@gmail.com, catalin.negru@cs.pub.ro
***Computer Science Department, University Politehnica of Bucharest, Bucharest, Romania,*
National Institute for Research and Development in Informatics (ICI), Bucharest, Romania (e-mail:florin.pop@cs.pub.ro)

Abstract: The Cloud is today one of the most popular technologies in the field of Information Technology. When dealing with constraints related to a too low number of resources that are offered by a single Cloud service provider the implementation of a Multi-Cloud Federation intervenes. In the case of a Multi-Cloud Federation, Cloud service customers might interact with other Cloud service providers that offer the required number of resources. Other possibility is when the Cloud service provider (CSP) rents resources from other Cloud service providers. There also might be a composition of the two strategies. When dealing with Multi-Cloud Federation in the case of Cloud service providers it is critical to not violate the Service Level Agreement (SLA) contract that is in place between the Cloud service provider that uses various Cloud Service Infrastructures and the Cloud service customers (CSCs). In this article are discussed those SLA parameters and the security and trust ideas behind the Multi-Cloud Federation. We also make an analysis of the capacity variation of the Multi-Cloud Federation in different conditions.

Keywords: Multi-Cloud Federation, SLA parameters, SLA constraints, Authentication, Authorization.

1. INTRODUCTION, RELATED WORK AND NOVELTY OF THIS WORK

In today's computing world Cloud technologies are more and more utilized. Cloud Customers become interested in Cloud technologies that are useful in different situations. The Cloud infrastructures rely on virtual machines and multiple resources shared on the same virtual or physical machine(s), on multiple virtual or physical machine(s), or even between two or multiple clouds.

When dealing with constraints related to a too low number of resources that are offered by a single Cloud service provider, the Cloud service customers might interact with other Cloud service providers that offer the required number of resources. Other possibility is when the Cloud service provider rents resources from other Cloud service providers. There also might be a composition of the two strategies.

When dealing with Multi-Cloud Federation in the case of Cloud service providers it is critical to not violate the SLA contract that is in place between the Cloud service provider that uses various Cloud Service Infrastructures and the Cloud service customers.

Cloud federation is the subject of recent research efforts. In (Sette et al., 2017) it is defined an "Authorization Policy Federation" for heterogeneous cloud accounts. The purpose is to allow the Multi-Cloud service customers to have the same access policy across multiple Clouds. (Pustchi, N., et al.) show how to share the Cloud resources across homogeneous Multi-clouds. (Ahmed, U., et al.) discuss the necessity of trust models in Multi-Cloud Federation and the authors present and analyze the "Trust Management Systems (TMS)" proposed in literature to address the Multi-Cloud Federation. In (Hong et al., 2019) an overview of the newest technologies that are encountered in Multi-Cloud Computing is presented. Another interesting aspect of Multi-Cloud Federation is presented in (Dreibholz et al., 2019). In this work it is analyzed how to use real world Cloud applications that are deployed in a Multi-Cloud Federation. In all these works it is shown that Multi-Cloud Federation is a solution for Cloud resource sharing and for Cloud computing.

The motivation behind this article is to address the SLA aspects that need to be considered in the case of Multi-Cloud Federation. When dealing with an interaction between a Cloud service customer and one or more Cloud service providers a SLA contract is needed. The main contributions of this paper, which reflect the novelty of this work are as follows:

- it discusses the necessity of authentication and authorization in multi-Cloud environments and its relationship with the defined SLA monitoring module;
- it discusses the SLA in Multi-Cloud Federations and what are the different SLA contracts that need to be described in such environments;
- it discusses the SLA parameters that need to be defined when talking about Multi-Cloud Federations;
- an analysis based on Markov chains and queuing theory that allow us to estimate the blocking probability according with SLA restrictions and the number of jobs waiting in the queue to be processed by a service.

The necessity of an SLA when dealing with Multiple Cloud service providers is described from the architectural point of

view of an SLA in Multi-Cloud Federations. Also, the parameters that need to be described are presented in this work.

This paper is organized as follows: Section 1 presents an introduction regarding the SLA constraints in multi-cloud environments, the related work and the novelty of this work, Section 2 discusses the reasons behind federation of clouds environments, Section 3 describes the authentication and authorization mechanisms used in multi-clouds, Section 4 presents the parameters that must be taken into consideration regarding SLA in federated clouds. Finally, Section 5 presents conclusions and future work.

2. REASONS BEHIND FEDERATION OF CLOUDS

The reasons behind federation of Clouds (Liaqat et al., 2017) are multiple and they can be explained based on the SLA characteristics, parameters and case when there SLA violations occur (see Table 1). In order to satisfy the Cloud Service Customer requirements, the Cloud service providers can use multiple Cloud infrastructures, some of them rented from other Cloud service providers (Fig. 1 b), Fig. 1 c)). Another situation can be encountered when the Cloud Service Customers use the Cloud infrastructures of various Cloud service providers directly with various purposes (see Fig. 1 a), Fig. 1 c)). Both situations are based on the idea of Cloud Federation. Cloud Federation is encountered in the case of multiple Clouds that interconnect with each other. For example, one of the reasons behind this interconnection is the fact that single Clouds Service Providers have a finite capacity. To increase this capacity, the Cloud service providers use Cloud federations or Federated Clouds. All the reasons behind Cloud Federation are explained in detail in Table 1.

Table 1. Reasons and SLA aspects of the federated clouds.

Reasons	Motivation	SLA
1. Sharing (Xu et al., 2017)	There is a need for sharing between different Cloud service providers because of different technical characteristics(for example data is shared between different Cloud service providers for faster access) and different trust (a Cloud Service Customer trusts two or more certain Cloud service providers more and thus shares between them sensible data) and policies aspects required by the Cloud Service Customers (for example the shared data contains sensible content and it must be shared only with some given Cloud service providers)	In the SLA contract must be specified that the sharing must take place because of several reasons (for example those described in the motivation column) and the SLA parameters that describe the need for the sharing process must be specified in detail. SLA violations describe the situations when the technical capabilities of the Cloud service providers that compose the Federated Cloud are violated (for example the access to the data takes more than a given threshold). Another situation is when the trust is necessary meaning that certain Cloud service providers are trusted to do some operations (for example on the shared data). In this case the trust in some Cloud service providers must be defined in the SLA contract. In the case of different access policies to the data various SLA parameters must be defined such as when SLA violations that refer to the Cloud Service Customers organizational policies are encountered.
2. Fault tolerance (Garraghan et al., 2011)	The fault tolerance aspect refers to replications (of the data for example) with the purpose of assuring that the Cloud service is failsafe. More than one Cloud service providers must be considering when trying to ensure fault tolerance.	In this case the SLA must define what and how many Cloud service providers must be activated in parallel and that assure the fault tolerance of the overall Cloud service. The number and type of Cloud service providers must be agreed based on the requests of the Cloud Service Customers and the Cloud service providers.
3. Improved QoS (Garraghan et al., 2011)	The QoS can be improved when there are different geographically distributed Cloud service providers and the closest to a certain Cloud service customer (to reduce the network latency for example) should be found.	In the SLA must be specified which is the maximum accepted latency such that the Cloud service providers know if they can offer such a maximum latency threshold.
4. Cost efficiency (Taleb et al., 2013)	By having different possible Cloud service providers, the cheapest one or more than one and that suits our necessities should be chosen.	The SLA must contain operational costs for each Cloud service provider for the Cloud service customer to choose the best Cloud service provider(s).

5. Reducing SLA violations (Ahmed et al., 2019)	In the case that SLA violations occur, the Cloud service provider can rent additional resources from other Cloud service providers.	Various SLA parameters must be defined with the purpose of capturing the various SLA violations.
6. Provider Independence (Esposito et al., 2013)	By choosing more than one Cloud service providers the Cloud service customers is not dependent upon a single Cloud service provider.	The Cloud service customer negotiates several SLAs, one SLA with each Cloud service provider.
7. Contract ending (Liaqat et al., 2017)	If there are more than one operational contract between the Cloud service customer and the Cloud service providers and if one contract ends, then another Cloud service provider can be chosen such that the service does not stop.	In this case several SLA contracts must be in place between each Cloud service provider and the Cloud service customer. If a contract between the Cloud service customer and one Cloud service provider ends another SLA contract between another Cloud service provider and the Cloud service customer is in place.

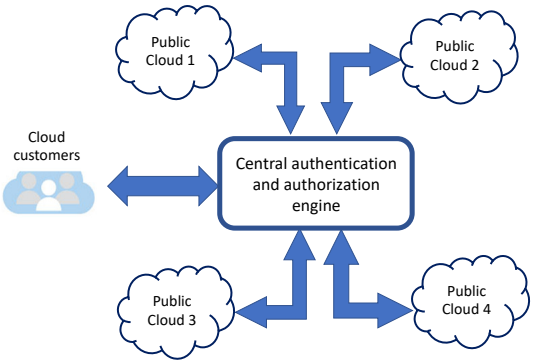

Fig. 1. a). Centralized Multi-Cloud Federation. A central authentication and authorization engine are used to allow access to Clouds.

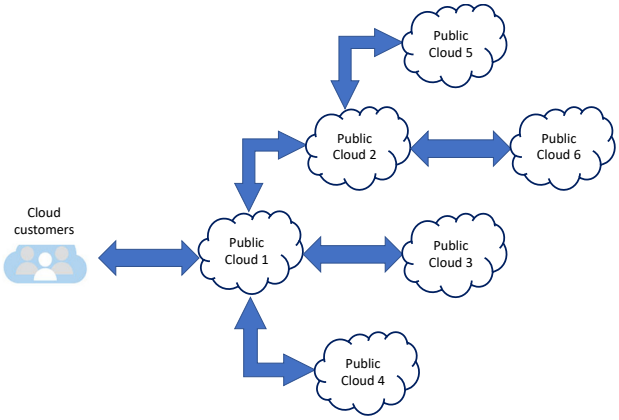

Fig. 1. b). Hierarchical Multi-Cloud Federation. The Cloud customer(s) interact(s) with only one Cloud provider that might collaborate with one or more Cloud providers.

There can be various architectures of Federated Cloud (see Fig. 1 a), Fig. 1 b), Fig. 1 c)). The Cloud Federation is becoming a more and more popular both for the Cloud service customers and for the Cloud service providers.

In the case of multiple Clouds one of the main concerns and constraints are those related to Service Level Agreement violations generated by the implementation of the multi-Cloud as shared resources. In order to function correctly the Federated Cloud infrastructure must define correctly various parameters and those parameters cannot be violated.

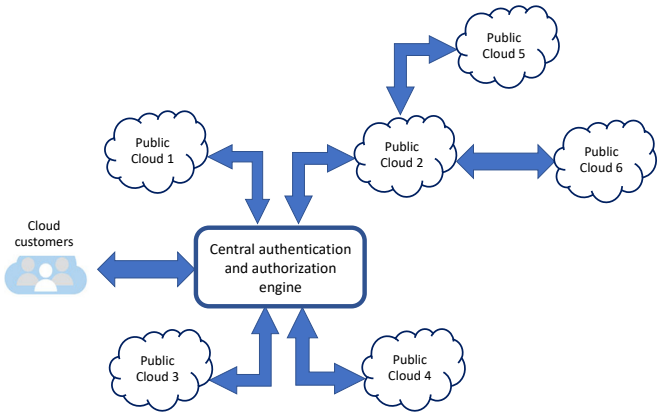

Fig. 1. c). Heterogeneous Multi-Cloud Federation. The Cloud customer(s) interact(s) with a central authentication and authorization engine might collaborate with one or more Cloud providers.

The first Federated Cloud infrastructure element that needs to be implemented is the one that ensures the security and trust aspects of the Cloud infrastructure. The security related aspects are discussed as follows.

Centralized Multi-Cloud Federation. In the first figure, Fig. 1 a) there exists a central authentication and authorization engine with the purpose of integrating the Cloud Service infrastructures that compose the Cloud Federation. Also, the access to the Federated Cloud resources is made individually by interacting with each Cloud service provider individual entity. When designing such a central authentication and authorization engine various aspects must be taken into consideration such as (Esposito et al., 2016):

- Single Sign On such that the Cloud service customer authenticates only once to the Central Authentication and

Authorization Engine (CAAE) and the CAAE controls the encryption and privacy of data and does not allow the Cloud service provider(s) to control the encryption and privacy of data (Esposito et al.; Togan et al., 2015);

- data geolocation;
- when designing a Federated Cloud, the Cloud service providers location is also necessary.

The Single Sign-on is useful when having to deal with various Clouds that are used by the same Cloud service customer (Ghazizadeh et al., 2012).

The data geolocation (Esposito et al.) is important when discussing about knowing the location of data. This must be taken into consideration by the Cloud service customers in order to know if the legislation is broken (Esposito et al.) (for example in the case that an USA-based Cloud service provider has access to the EU-based data Cloud service customer but the legislation forbids such access).

It is necessary to know the Cloud service providers location because of data geolocation (the data must be stored in a Cloud infrastructure that meets the location legislation awareness policies or (another example) if the processing of the Cloud Information must take place in the same location that is the same as the one of the Cloud service customer).

Hierarchical Multi-Cloud Federation. Another situation described in Fig. 1 b) the Cloud service customers design a Service Level Agreement by discussing with only one Cloud service provider. In this case of interest is the interaction between different Cloud service providers that are used for their resources by the main Cloud service provider (that owns the Public Cloud 1 – see Fig. 1 b). In this case there must exist an SLA between the various Cloud service providers that own the various Cloud infrastructures. The SLA between the main Cloud service provider (the one that interacts with the Cloud service customers) must have the same parameters and values for these parameters with the ones that exist between the various Cloud service providers. In this case the negotiation of the SLA takes place between:

1. the Cloud service customers and the main Cloud service provider
2. the main Cloud service provider and the rest of the Cloud service providers

In other words, the security and trust aspects must be the same between the various Cloud service providers with those between the Cloud service customers and the main Cloud service provider. For the Cloud service customers, the interaction with the main Cloud service provider is a usual one between a single Cloud service provider and Cloud service customers.

In the second case there is a need to implement secure and trusted inter-Cloud Interaction Engine.

Heterogeneous Multi-Cloud Federation. In the case of heterogeneous Multi-Cloud Federation (Fig. 1 c)) both the Central Authentication and Authorization Engine (CAAE) and the secure and trusted inter-Cloud Interaction Engine must be implemented.

Next, the privacy of the data aspects in the Federated Cloud environments and the trust aspects that intervene in the interaction between the Cloud service customers and the Federated Cloud are discussed. In a Federated Cloud it is crucial to ensure the data privacy in certain cases (for example when working with private data such as private information – for example medical data). This generates the need for a global contract (a global SLA contract too) between the Federated Cloud and the Cloud service customers. Additionally, a trust relation must be in place both between the Cloud service customers and the Federated Cloud and between the entities that form the Federated Cloud. The trust relation between the Cloud service customers and the Federated Cloud must be specified in the SLA contract between the two entities. The trust between the entities of the Federated Cloud must be specified in the SLA between the Federated Cloud entities. One of the most important ideas when discussing about the Federated Cloud infrastructure is that each Cloud service provider entity should be able to administer and control its resources. In addition, in case of the Hierarchical Multi-Cloud Federation (and in the Heterogeneous Multi-Cloud Federation) the main Cloud service provider must be able to request resources as negotiated in the SLA contract and keep track of the resources that it requests.

Access to resources in the case of Centralized Multi-Cloud Federation, in the case Hierarchical Multi-Cloud Federation and in case of the Hierarchical Multi-Cloud Federation from the point of view of the Cloud service customers must be transparent, meaning that the Cloud service customers must have access to each the Federated Cloud resources that are published for use.

Furthermore, Cloud service customers must be trusted when accessing different Federated Cloud resources and, in the case of Hierarchical Multi-Cloud Federation and in case of the Heterogeneous Multi-Cloud Federation the Main Cloud service provider must have access to the resources of the other Cloud service providers as stipulated in the SLA agreement.

3. CENTRAL AUTHENTICATION AND AUTHORIZATION ENGINE

The federated collaboration between Clouds (Fig. 1 a), Fig. 1 b), Fig. 1 c)) implies a high degree of inter-dependence and trust among Clouds. In the case of Fig. 1 a) and Fig. 1 c) the Central Authentication and Authorization Engine (CAAE) should be based on a global meta-policy as discussed in (see Fig. 2) (Almutairi et al., 2012). This global meta-policy should contain and utilize the access and control policies for each of the Cloud entities (providers).

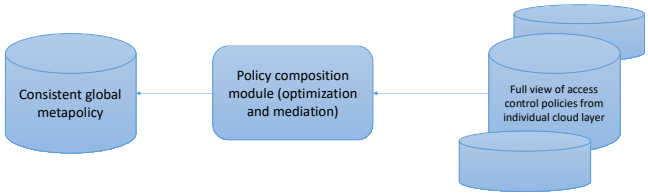

Fig. 2. Centralized Multi-Cloud Federation. A central authentication and authorization engine are used to allow access to Clouds.

In addition, the central authentication and authorization engine should enable secure inter-operation between cloud customers and the heterogeneous Cloud providers by using the following principles (Gong and Qian, 1996):

- **Autonomy Principle** – this implies that if a Cloud customer can access a given Cloud system, it must also have the same permissions under secure federation of Clouds.
- **Security Principle** - if a Cloud customer cannot access a given Cloud system then it must not have access to that Cloud system under secure federation of Clouds.

The central authentication (Korac, 2017) and authorization (Morariu et al., 2013) engine has the purpose of determining if a **subject** (Cloud service customer) has the **privilege** to perform a given action over the controlled **object** (resource of the Multi-Cloud Federation Architecture (MCFA)) (Calero, J. M. A., et al.). A tuple formed by three terms (**Subject, Privilege, Object**) can be used.

To deal with different Cloud service providers (in the case of the MCFA architecture), the 3 terms tuple must be extended by using a 4-tuple (**Issuer, Subject, Privilege, Object**). In this 4-tuple the **Issuer** is a Cloud service provider which uses the central authentication and authorization engine. Additionally, the authentication and authorization engine can be extended by adding another field to the 4-tuple, **Interface**, which represents the interpretation of the **Object**, thus by having a 5-tuple (**Issuer, Subject, Privilege, Interface, Object**). Then, this 5-tuple may be interpreted as: The **Issuer** says that the **Subject** has the **Privilege** to perform a given action over the **Object** associated to the type **Interface**. As an example, the 5-tuple (Florin, George, Read, CloudStorage, \root\) may be interpreted as: Florin says that George can Read the \root\ folder associated to the CloudStorage service. It is possible that for a **Privilege** to enable multiple actions. For example, the Write might enable the actions Delete and Update.

To implement the central authentication and authorization engine the access control mechanism is described above. It can be seen that the access mechanism is a RBAC based one (Shafiq et al., 2005; Mohammad et al., 2002). The basic idea behind the RBAC access control mechanism is that to access a given resource the user needs to have a certain role. The access decision is taken by the central authentication and authorization engine while considering the roles defined for a given user and the permissions for that role.

Additionally, when discussing about Hierarchical Multi-Cloud Federation a Cloud provider that is lower in the hierarchy (e.g. the Cloud provider 2) must trust the Cloud provider that is upper in the hierarchy (e.g. the Cloud provider 1).

In the case of a Heterogeneous Multi-Cloud Federation both the trust model and the central authentication and authorization engine come into place.

Next, some of the features of the Central Authentication and Authorization Engine (CAAE) will be discussed. Three different monitoring modules were identified, and they must be part of the:

- **Cloud Service Customer Behavior Monitoring Module (CSCBMM)**: The CSCBMM has the purpose of monitoring the actions of the Cloud service customers during the interaction with the Federated Cloud Environment. During this interaction the Cloud service customers can perform legal actions or any malicious activities. Thus, the CSCBMM monitors and record all the actions performed by the Cloud service customers.
- **SLA Monitoring Module (SLAMM)**: This module has the purpose of checking whether the parameters of the SLA between the Multi-Cloud Federation and the Cloud service customers are violated or not. The monitoring process must be a continuous one (meaning that when a Cloud service customer uses the Multi-Cloud Federation all the vital SLA parameters should be monitored).
- **Trust Evaluation Module (TEM)**: This module has the purpose of collecting the information about the interaction between the Cloud providers that are part of the Hierarchical Multi-Cloud Federation or to the Cloud service providers that are part of the Heterogeneous Multi-Cloud Federation to see if the trust relationship between Cloud service providers is correct.

4. SLA IN FEDERATED MULTI-CLOUD

Various SLA parameters can be discussed when dealing with Federated Multi-Cloud. The SLA contract in the case of Multi-Cloud Federation has three different organizations and definitions (see Fig. 3 a), 3 b), 3 c)). In the case of Centralized Multi-Cloud Federation, it can be seen that the Cloud service customers have the possibility to negotiate individual contracts with each of the Cloud service providers).

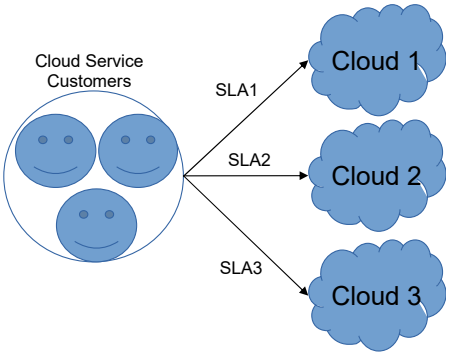

Fig. 3. a). SLA in the case of Centralized Multi-Cloud Federation.

In the case of Hierarchical Multi-Cloud Federation, the SLA contract is negotiated between the Cloud service customer with the main Cloud service provider and the other Cloud service providers (the secondary ones) negotiate the same SLA contract with the main Cloud service provider (Cloud 1 in Fig. 3. b)).

In the case of Heterogeneous Multi-Cloud Federation, it can be seen that there is a mix between the two different SLA contract negotiation strategies.

The first SLA parameter is the **Maximum discovery time**. This parameter refers to the time necessary for a Cloud service

provider to expose its resources and how much time it takes to the other Cloud service providers in the Multi-Cloud Federation to discover these resources. When dealing with Multi-Cloud Federation the resource discovery is important to handle the time to discover a resource that has a certain geographical distance from the location of the Cloud service customer and some physical (in terms of technology) distance. In addition, the Cloud service providers Inter-Clouds communication costs must be taken into consideration.

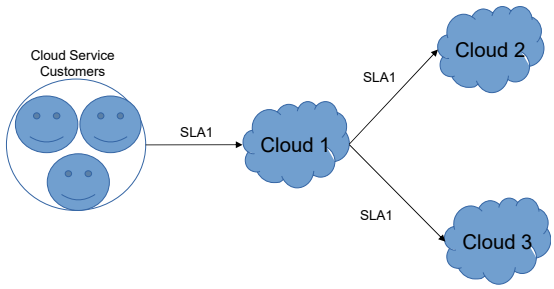

Fig. 3. b). SLA in the case of Hierarchical Multi-Cloud Federation.

In the case of the Centralized Multi-Cloud Federation the Maximum discovery time must be considered in of each of the used Clouds by the Cloud service customer (in the case of Fig. 3 a) SLA_1, SLA_2 and SLA_3 are defined).

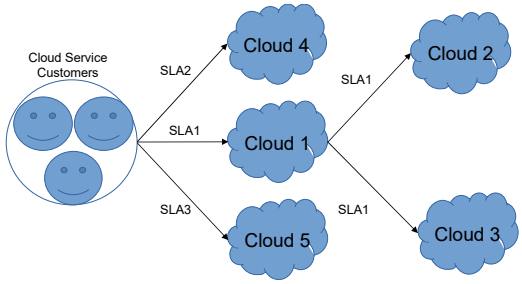

Fig. 3. c). SLA in the case of Heterogeneous Multi-Cloud Federation.

In the case of Hierarchical Multi-Cloud Federation, the Maximum discovery time must be the same and it must be defined in the SLA_1.

In the case of Heterogeneous Multi-Cloud Federation, the Maximum discovery time is a combination of SLAs.

Another SLA parameter that can be considered in the case of Multi-Cloud Federation is the High Computation Overhead. In this case Multi-Clouds with some computational capabilities are encountered. The High Computation Overhead is defined when trying to compute something on multiple Clouds with different computational capacities. Thus, in this case a computational overhead can occur. This SLA parameter is necessary from the computational point of view.

For example, various sub-parameters can be defined in this case:

- The **Percent of Correctly Executed Cloud Tasks on the Multi-Cloud Federation Architecture (PCECT_MCFA)** defines the percent of tasks that are executed correctly by the Multi-Cloud Federation Architecture (MCFA). The $PCECT_{MCFA}$ parameter can be defined as being the Total Number of Tasks Executed Correctly divided by the Total Number of Cloud Tasks:

$$PCECT_{MCFA} = \frac{\text{Total No.of Tasks Executed Correctly}}{\text{Total No.of Cloud Tasks}} \quad (1)$$

A task can be executed incorrectly on the MCFA for several reasons (other reasons than those specified in [12]):

i. The scheduler used by the MCFA is not able to schedule the tasks on the Cloud infrastructure that has the capacity (for example CPU capacity in MIPS or the memory capacity in GB) to run the tasks.
ii. The scheduler used by the MCFA schedules some tasks on the Cloud infrastructure that does not have the rights to execute the tasks, thus decreasing the ***PCECT_MCFA***.
iii. The scheduler used by the MCFA may not be able to schedule correctly concurrent tasks, thus concurrent access to the resources can generate a lack of necessary Cloud resources.
iv. The partial or total stoppage of one or more Cloud infrastructures that form the MCFA. In the case of a MCFA a CSP can stop functioning. In this case all the tasks that are sent by the Cloud service customers (CSCs) to be executed on that CSP (or on those CSP) cannot be scheduled.
v. Too big latency of the MCFA network (the time of the transmission through the MCFA network of the parameters of a task or of the execution results is too big). In the case of MCFA it is possible that some parameters related to the network latency are violated because of several reasons such as: physical infrastructure reasons, network software reasons, etc.
vi. The scheduler used by the MCFA cannot schedule the tasks in a given frame of time thus one or more tasks execution time expires. The execution time of the tasks that are sent to be executed can expire due to the lack of resources, too big network latency, or the MCFA scheduler is not implemented correctly when sending the tasks to be executed.
vii. The number of tasks send to be executed exceeds the processing capacity of the MCFA. In this situation the MCFA must be modified in order to acquire more resources (new Cloud service providers can be added to the MCFA).

- Another SLA Cloud parameter is the Percentage of Disponible Resources **(PDR)**. This parameter can be expressed as the percentage of disponible resources during the functioning of the MCFA. This parameter can be expressed as the ratio between the Number of Available Resources and the Total Number of Resources during a certain period. The PDR can be defined with the following formula:

$$PDR_{MCFA}(time) = \frac{\text{Number of Available Resources}}{\text{Total Number of Resources}} \quad (2)$$

When defining this parameter, it must be considered that some of the MCFA resources might not be available. The reason because the resources are not available can be the following:

i. some of the MCFA service resources are not available to a certain Cloud service customer (CSC);
ii. some MCFA resources can be stopped;
iii. the requested resources are dealing with other requests possibly from other clients.
iv. in the case of the MCFA its resources may not be available because of the concurrent access to the resources;

- Furthermore, another parameter is given by the Maximum Time of Incorrect Functioning **(TIF)** of the MCFA Service or as the Maximum Percentage of Incorrect Functioning **(PIF_{MCFA})** of the MCFA Service.

$$PIF_{MCFA} = \frac{\text{Time of Incorrect Functioning}}{\text{Total Functioning Time}} \quad (3)$$

This parameter is based on the idea that the MCFA service does not function correctly or that the MCFA service is stopped. This parameter can be greater than zero for different reasons:

i. concurrent access to the MCFA resources
ii. Cyber-attacks on the MCFA Service
iii. the MCFA failure or the stopping of the MCFA Service
iv. maintenance work on the MCFA
v. another situation when the MCFA Service might not function correctly is because of malicious Cloud service customer(s) that incur bad functioning of the entire MCFA Service.

- The **Total Time of Execution/Transmission** parameter is expressed as the time frame from when a task is sent to execution and the result is sent back to the Cloud service customer. This parameter is taken into consideration because is necessary when discussing about the scheduling of the tasks This parameter is necessary also in the case of a MT-CSP because the Total Time of Execution/Transmission can be increased when dealing with multiple tenants that run concurrent application on the same Cloud service provider.

The parameter called **Maximum response time** refers to the time period from when a request is sent by a Cloud service customer to the Multi-Cloud Federation Cloud service providers and the time the response is received. In this case the **Maximum response time** can be associated with the **Total Time of Execution/Transmission.**

The parameters called **Cloud Service Disponibility (CSD)** refers to the number of time units in which the Multi-Cloud Federation Architecture Service (MCFAS) is disponible in average or the Percentage in which the MCFAS is Disponible given the Total Time of Functioning of the MCFAS. The percentage parameter is given by the ratio between the Total Time of Correct Functioning (TTCF) and the Total Time of Functioning (TTF).

$$PDS_{MCFA} = \frac{\text{Total Time of Correct Functioning}}{\text{Total Time of Functioning}} \quad (4)$$

Another important parameter is the **Resource relocation overhead**. In this case the overhead of relocating a certain resource on another Cloud service provider in terms of variations of the technological needs for the Cloud service providers should be considered and the execution times of certain tasks on the new Cloud service providers resources.

The **costs** represent another important SLA parameter. Various costs can be considered such as:

- **varied network traffic cost** (these can be minimized by reducing the networking costs by optimizing the routing process);
- **high operational costs**; for example, to reduce the resource allocation costs in terms of VMs to avoid SLA violations under SLA constraints was studied in the case of SLA constraints-see (Iordache et al., 2017) and Cloud service provider with multiple tenants - see (Iordache, 2019);
- **costs related to the budget of the Cloud service customer** can be minimized by negotiating with several Cloud service providers;
- **backup costs.** This cost is expressed in number of resources necessary to do backup of a Cloud service customer application. In this case the **cost of storage** should be considered;
- **replication costs.** This cost is important when optimizing the allocation cost of resources if there is a need to schedule various tasks on various resources;
- **costs of maintenance and deployment.** These costs must be considered when dealing with the maintenance of the Cloud Services offered and with the deployment of applications from the Cloud service customers.

Other parameters consider the implemented functionalities from different perspectives such as the **consumed energy** of the Cloud System. (the Cloud service providers can be interested in diminishing the consumed energy) (Negru et al., 2013).

Case study. A good case study for assuring SLA in multi clouds environments is represented by the satellite images processing applications. This case study was presented in detail in our previous paper (Ilie et al., 2019). These applications are built on multiple components developed through specialized libraries for manipulating geo-reference images, performing numerical calculations on matrix structures, or providing efficient parallelization mechanisms in a distributed way. The implementation is based on master-worker programming model, where the master reads the data, calculate the size of the chunks of the images that each process receives and transmits data. Worker processes process data and send the results back to the master process, which aggregate the partial results. Usually, satellite images come from

different sources and are stored on different locations at various Cloud service providers. Thus, in order to process the data, a multi-cloud environment for efficiency and cost optimization must be designed. When running algorithms on massive datasets the I/O part could become a bottleneck, so computations near data must be performed. This means that different phases of the algorithms must be coordinate over multiple clouds.

The metrics proposed in Section 4 are useful for designing an SLA with the purpose of executing the algorithms efficiently. For example, the SLA parameter **costs** of the Multi-Cloud Federation is very important when negotiating the SLA for a satellite images processing Multi-Cloud Federation based application.

Another parameter that is important is the disponibility of the Multi-Cloud infrastructure. If the processing of images needs to be done in real-time then the disponibility of the Multi-Cloud must be near 100% when dealing with new images that need to be analyzed. The same can be said about the **Percent of Correctly Executed Cloud Tasks on the Multi-Cloud Federation Architecture** (if the algorithms need to be run in real time then they must function without fault).

Additionally, the **Total Time of Execution** needs to be minimized when dealing with algorithms that take a long period of time to be executed (image segmentation, risk analysis maps, etc.).

The **Percentage of Disponible Resources** must also be at a certain level because the processed images have certain requirements in terms of necessary resources.

In the case study presented it is interesting to discuss the scheduling of tasks in a Federated Multi-Cloud environment used to reduce the costs. In the case of satellite images processing applications, the images are stored on servers placed in different geographical locations. In this context it is useful to know where we need to execute the algorithms for image processing (for example to minimize costs).

If we want to send a number N of tasks to be scheduled on a Federated Multi-Cloud environment (that has three different Clouds) then we have to think about how many tasks are needed to be send to each Cloud from the Federated Multi-Cloud environment in order to minimize costs.

For example, if we want to minimize costs, we consider the scheduling of the N initial tasks on the three public Clouds. We can write $N=n_1+n_2+n_3$. In this sum n_1 represents the tasks are scheduled on Cloud 1, n_2 the tasks that are scheduled on Cloud 2 and n_3 the tasks that are scheduled on Cloud 3. The formula for the total costs when scheduling in this Federated Multi-Cloud environment is given by:

$$C = n_1*c_1 + n_2*c_2 + n_3*c_3 \quad (5)$$

where n_1, n_2, n_3 and c_1, c_2, c_3 are variable.

Table 2. Costs of scheduling the N number of tasks on the Federated Multi-Cloud environment (we used an abstraction for cost values that is useful only for comparison).

$N=n_1+n_2+n_3$	n_1	n_2	n_3	c_1	c_2	c_3	C
20	6	7	7	5	7	8	**135**
20	10	5	5	6	7	8	**135**
20	12	4	4	6	7	8	**132**
20	14	3	3	7	6	8	**140**
20	16	2	2	8	6	6	**152**
20	17	2	1	9	1	6	**161**
20	19	1	0	9	1	0	**172**

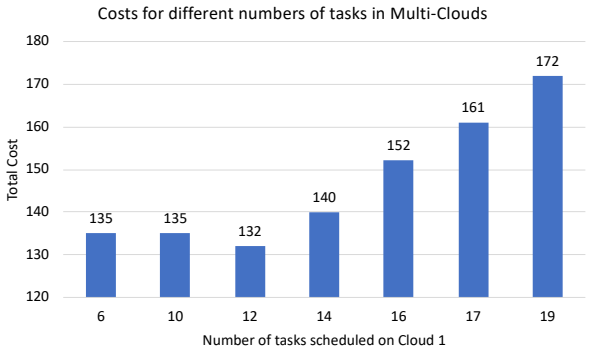

Fig. 4. Costs for different numbers of tasks in Multi Clouds.

In Table 2 we can see different values for n_1, c_1, n_2, c_2, n_3 and c_3. We can see that the costs increase with the number of tasks that should be scheduled. This example is based on the idea that the costs c_1, c_2, c_3 increase with the number of tasks allocated on a certain Cloud. This is the case in real time scenarios (for example see Amazon model[1]). In the given example the pricing of the execution of tasks varies with the number of tasks and with the geographical location. The idea is to minimize the total cost C, by obtaining an equilibrium in the sum C of its members.

We can conclude that the total cost can be minimized when its value is 132 (see Table 2 and Figure 4). Various configurations must be computed to achieve an optimal cost. The designer of the SLA contract must consider several configurations (like in Table 2) to minimize the total costs of the scheduling. Thus, to optimize a certain parameter (meaning that we minimize, or we maximize it) a process must be followed.

From the queueing theory (Harchol-Balter, 2013) we know that if we have a service on a public Cloud that have an average service rate of μ jobs/second (exponentially distributed) and an average arrival rate of λ jobs/second (Poisson processes) with $\lambda < \mu$ we can compute $\rho = \lambda/\mu$ as service utilization and it represents the fraction of time when the service is busy. If we have k identical services on the same Cloud provider, then the

[1] https://aws.amazon.com/swf/pricing/

utilization of each service becomes $\rho = \lambda/(k\mu)$ where $k\mu$ represents the total task processing capacity.

We can model a Federated Multi-Cloud environment having k services located in different Clouds. In the case of satellite image processing applications, we have a central point that works as a broker where the tasks arrive, and they are distributed to different services. We consider that between the tasks that arrive we do not know the interarrival times. Additionally, the service times of each server is not known apriori. More, k homogeneous servers share a pool of tasks that is common for the entire environment. This type of service from a Cloud has a processing capacity of k jobs. So, we face with an M/M/k (or M/M/k/k) system.

We are interested in two different parameters:

1. the blocking probability P_{block} (Harchol-Balter, 2013), which denotes the ratio of jobs that are lost because the capacity of the Federated Multi-Cloud environment is too small; this probability represents the possibility that when a new task comes into the system finds all k servers busy.

2. the probability P_Q that a new job that arrives into the system must be queued. This means that a new job encounters more than k jobs in the environment.

According with the system model P_Q is (Harchol-Balter, 2013):

$$P_Q = \frac{(k\rho)^k}{k!(1-\rho)}\left(\frac{(k\rho)^k}{k!(1-\rho)} + \sum_{i=0}^{k-1}\frac{(k\rho)^i}{i!}\right)^{-1} \quad (6)$$

Based on (6), we can compute

$$P_{block} = \frac{(1-\rho)P_Q}{1-\rho P_Q} \quad (7)$$

These probabilities express how to design the Federated Multi-Cloud environment to share a specific capacity. For example, if P_{block} is too big, we have the risk to drop many tasks, so we must increase the number of servers because we do not want to break the SLA rules. In the case of P_Q, we must increase the size of the queue or add more servers to the environment.

If we plot the graph (see Figure 5) of P_Q, we can see that it varies with ρ and with k and the smallest values of P_Q are when k = 128 and ρ = 0,95 (see Table 3).

Table 3. P_Q for μ = 1 on multi-server environment.

	P_Q							
ρ (μ = 1)	k=1	k=2	k=4	k=8	k=16	k=32	k=64	k=128
0,95	**0,95**	**0,926**	**0,891**	**0,844**	**0,78**	**0,696**	**0,588**	**0,458**
0,96	0,96	0,94	0,913	0,874	0,822	0,751	0,659	**0,543**
0,97	0,97	0,955	0,934	0,905	0,864	0,809	0,736	**0,640**
0,98	0,98	0,97	0,956	0,936	0,908	0,87	0,818	**0,748**
0,99	0,99	0,985	0,978	0,968	0,954	0,934	0,906	**0,868**

If we plot the graph (see Figure 6) of P_{block} we can see that P_{block} varies slightly with ρ and more with k and the smallest values of P_{block} are when k = 128 and ρ = 0,95 (see Table 4), which is the same results like in the case of P_Q.

Based on the previous results we represented the graph with ρ = 0.95 (see Figure 7) where the percentage of late customers represent P_Q, percentage of lost tasks represents P_{block} and we also computed the medium waiting time of late customers. We can see that the medium waiting time of late customers decreases with the k parameter as it does the percentage of late customers and the percentage of lost tasks.

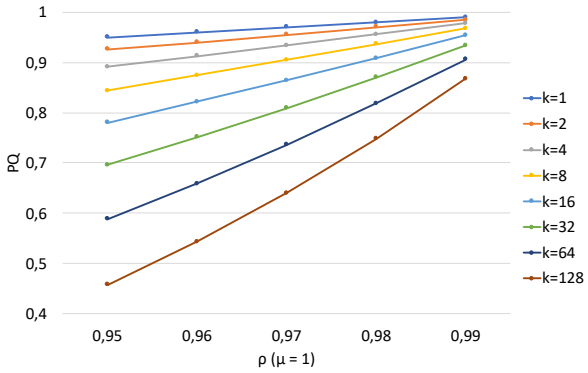

Fig. 5. P_Q for μ = 1 on multi-server environment.

Table 4. P_{block} for μ = 1 on multi-server environment.

	P_{block}							
ρ (μ = 1)	k=1	k=2	k=4	k=8	k=16	k=32	k=64	k=128
0,95	**0,487**	**0,384**	**0,291**	**0,213**	**0,151**	**0,103**	**0,067**	**0,041**
0,96	0,490	0,387	0,295	0,218	0,156	0,108	0,072	**0,045**
0,97	0,492	0,390	0,299	0,222	0,161	0,113	0,077	**0,051**
0,98	0,495	0,394	0,303	0,227	0,165	0,118	0,082	**0,056**
0,99	0,497	0,397	0,307	0,231	0,17	0,123	0,088	**0,062**

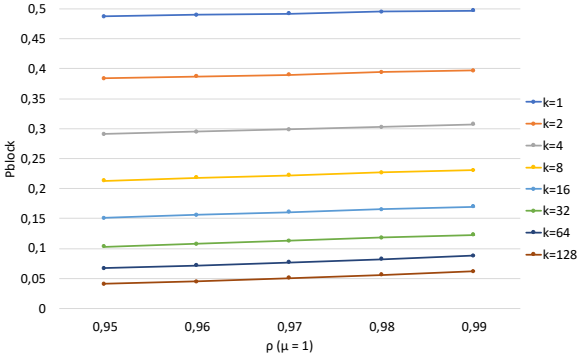

Fig. 6. P_{block} for μ = 1 on multi-server environment.

The average waiting time for late customers (tasks) is computed as follow:

$$E[T_Q] = \frac{1}{k\mu(1-\rho)} \quad (8)$$

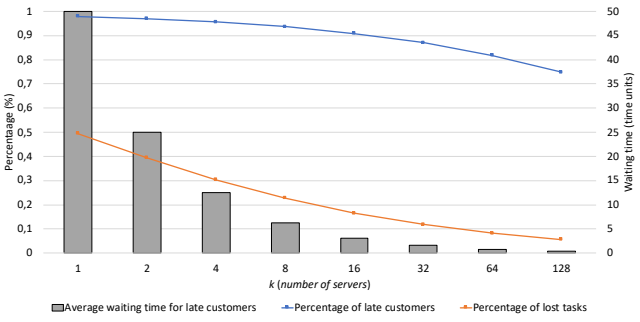

Fig. 7. Analysis of late customers, lost task and waiting time (ρ=0,95, μ=1).

We can conclude that the system offers a high scalability because the medium waiting time of late customers becomes very small for large values of *k*.

6. CONCLUSIONS AND FUTURE WORK

In this article it is motivated why there exists a need Multi-Cloud Federations. A mechanism that explains the process of authentication and authorization in a Multi–Cloud Federation is also presented. There are also discussed the parameters that are part of an SLA in the case of Federated Multi-Cloud environments. The case studies of the article suggest possible implementations of the SLA in real scenarios.

The work presented can be useful when doing research in the area of SLA in Cloud more specifically when attempting to design and model an SLA in real Multi-Cloud Federations.

As future work the implementation of the scheduling engine and under SLA constraints and optimize the SLA parameters in a real Federated Multi-Cloud scenario is planned. It is necessary to consider various SLA parameters that are in place in a Federated Multi-Cloud with the purpose of implementing scheduling and detecting SLA violations in such a real environment.

ACKNOWLEDGMENT

This work has been funded by University Politehnica of Bucharest, through the "ARUT Grants" Program, UPB – GNaC. Identifier: GNaC 2018, Contract: 16/06.02.2019, RM-CYBERSEC. The research presented in this paper is supported by the following projects: NETIO ForestMon (53/05.09.2016, SMIS2014+ 105976) and ROBIN (PN-III-P1-1.2-PCCDI-2017-0734). This publication was supported by a grant of the Ministry of Innovation and Research, UEFISCDI, project number PN-III-P2-2.1-SOL-2016-03-0046 (SPERO) within PNCDI III.

We would like to thank the reviewers for their time and expertise, constructive comments and valuable insight.

REFERENCES

Ahmed, U., Raza, I., & Hussain, S. A. (2019). Trust Evaluation in Cross-Cloud Federation: Survey and Requirement Analysis., pp. 1-37 *ACM Computing Surveys* (CSUR), 52(1).

Almutairi, A., Sarfraz, M., Basalamah, S., Aref, W., & Ghafoor, A. (2012). A distributed access control architecture for cloud computing, pp. 36-44. *IEEE software*, IEEE, 29(2).

Calero, J.M.A., Edwards N., Kirschnick J., Wilcock L., Wray M., (2010). Toward a multi-tenancy authorization system for cloud services, pp. 48–55, *Security Privacy*, IEEE.

Dreibholz, T., Mazumdar, S., Zahid, F., Taherkordi, A., & Gran, E. G. (2019). Mobile edge as part of the multi-cloud ecosystem: a performance study., pp. 59-66. In 2019 27th *Euromicro International Conference on Parallel, Distributed and Network-Based Processing* (PDP). IEEE.

Esposito, C., Castiglione, A., & Choo, K. K. R. (2016). Encryption-based solution for data sovereignty in federated clouds., pp. 12-17, *IEEE Cloud Computing*, IEEE, 3(1).

Esposito, C., Ficco, M., Palmieri, F., & Castiglione, A. (2013). Interconnecting federated clouds by using publish-subscribe service., pp. 887-903, *Cluster Computing*, 16(4).

Garraghan, P., Townend, P., & Xu, J. (2011). Byzantine fault-tolerance in federated cloud computing., pp. 280-285. In Proceedings of 2011 *IEEE 6th International Symposium on Service Oriented System* (SOSE). IEEE.

Ghazizadeh, E., Zamani, M., & Pashang, A. (2012). A survey on security issues of federated identity in the cloud computing. pp. 532-565, *In 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings.* IEEE.

Gong, L., & Qian X. (1996), Computational Issues in Secure Inter-operation. pp. 43-52. *IEEE Trans. Software Eng.,* vol. 22, no. 1.

Harchol-Balter, M. (2013). Performance modeling and design of computer systems: queueing theory in action. *Cambridge University Press.*

Hong, J., Dreibholz, T., Schenkel, J. A., & Hu, J. A. (2019). An Overview of Multi-cloud Computing., pp. 1055-1068. *In Workshops of the International Conference on Advanced Information Networking and Applications*. Springer, Cham.

Ilie, A. T., Filip, I. D., Postoaca, A. V., Negru, C., Pop, F., Stoica, A., & Serban, F. (2019, May). Faster and scalable parallel processing solution to remove visual obstacles from satellite imagery. In *2019 22nd International Conference on Control Systems and Computer Science (CSCS)* (pp. 194-201). IEEE.

Iordache, G., Paschke, A., Mocanu, M., & Negru, C. (2017). Service Level Agreement Characteristics of Monitoring Wireless Sensor Networks for Water Resource Management (SLAs4Water)., pp. 379-386, *In Studies in Informatics and Control,* 26(4).

Iordache, G., (2019). An analysis of Service Level Agreement parameters and scheduling in Multi-Tenant Cloud Systems, pp. 140-145, In 2019 22nd *International Conference on Control Systems and Computer Science* (CSCS). IEEE.

Korac, D. (2017). Design of fuzzy expert system for evaluation of contemporary user authentication methods intended for mobile devices., pp. 93-100, *Journal of Control Engineering and Applied Informatics,* 19(4).

Liaqat, M., Chang, V., Gani, A., Ab Hamid, S. H., Toseef, M., Shoaib, U., & Ali, R. L. (2017). Federated cloud resource management: Review and discussion., pp. 87-105, *Journal of Network and Computer Applications*, 77.

Mohammad A Al-Kahtani and Ravi Sandhu. (2002) A model for attribute-based user-role assignment., pp. 353–362, *In 18th Annual Computer Security Applications Conference*, IEEE.

Morariu, O., Morariu, C., & Borangiu, T. (2013). Transparent real time monitoring for multi-tenant j2ee applications., pp. 37-46*, Journal of Control Engineering and Applied Informatics,* 15(4).

Negru, C., Pop, F., Cristea, V., Bessisy, N., & Li, J. (2013, September). Energy efficient cloud storage service: key

issues and challenges. In *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies* (pp. 763-766). IEEE.

Pustchi, N., Patwa, F., & Sandhu, R. (2016). Multi cloud iaas with domain trust in openstack., pp. 121-123, In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy.*

Sette, I. S., Chadwick, D. W., & Ferraz, C. A. (2017). Authorization policy federation in heterogeneous multicloud environments., pp. 38-47, *IEEE Cloud Computing*, 4(4).

Shafiq, B., Joshi, J. B., Bertino, E., & Ghafoor, A. (2005). Secure interoperation in a multidomain environment employing RBAC policies., pp. 1557-1577, *IEEE Transactions on Knowledge & Data Engineering*, (11), IEEE.

Taleb, T., & Ksentini, A. (2013). Follow me cloud: interworking federated clouds and distributed mobile networks., pp. 12-19, *IEEE Network,* 27(5).

Togan, M., Morogan, L., & Plesca, C. (2015). Comparison-based applications for fully homomorphic encrypted data. *Proceedings of the Romanian Academy-series A: Mathematics, Physics, Technical Sciences, Information Science*, *16*, 329.

Xu, J., & Palanisamy, B. (2017). Cost-aware resource management for federated clouds using resource sharing contracts., pp. 238-245, In 2017 *IEEE 10th International Conference on Cloud Computing* (CLOUD). IEEE.