**A Cloud Forensic Readiness Model for Service Level Agreements Management**

Lucia De Marco[1, 2], Filomena Ferrucci[2], and M-Tahar Kechadi[1]
[1]School of Computer Science and Informatics, University College Dublin, Ireland
[2]Department of Management and Information Technology DISTRA MIT University of Salerno, Fisciano, Italy
lucia.de-marco@ucdconnect.ie
fferrucci@unisa.it
tahar.kechadi@ucd.ie

**Abstract:** Cloud computing is increasingly becoming a target of cyber-criminal attacks. Often the committed crimes violate the Service Level Agreement (SLA) contracts, which must be respected by all the involved parties. Cloud Forensics is a branch of Digital Forensic discipline dealing with crimes involving the Cloud. A manner for leveraging some of the attacks is the provisioning of a Forensic Readiness capability, by performing some activities before the crimes happen. In this paper we introduce a model aimed to represent the management of SLAs through a cloud system.

**Keywords**: Cloud Forensics Readiness, Service Level Agreements, Cloud Monitoring, SLA Model, SLA Management

## 1. Introduction

Cloud Forensics (Ruan et. al 2011) is the part of the Digital Forensic Science (Palmer 2011) dealing with the use of scientific approaches to investigate Cloud Computing platforms (Mell et. al 2011). Unfortunately, the current forensic techniques are not suitable for cloud-related crimes due to the challenges arisen by the Cloud physical structure and computing model (Birk et al. 2011, Mishra et al. 2012, NIST 2014, Reilly et. al 2011, Ruan et al. 2013, Reilly et al. 2010) (see Table 1). Cloud services are regulated by Service Level Agreements contracts (SLAs) (Mell et al. 2011), where some constraints among a cloud service provider and its customer(s) are detailed; the contracts are co-signed by the parties and they have legal validity in case of court litigation (Baset 2012, Patel et al. 2009). Because of cloud SLAs importance and contents (CSA 2013), they have been carefully monitored and verified: some attempts have been presented in literature (Czajkowski et al. 2002, Skene et al. 2007, Paschke et al. 2008, Unger et al. 2008, Ishakian et al. 2011, Gosh et al. 2012, EU 2013), also in the forensic community (De Marco et al. 2014), all devoted to better manage SLAs violations. We consider the implementation of a Digital Forensic Readiness capability (Tan 2001, Rowlingson 2004, De Marco et al. 2013) for the cloud as one of the best solutions to render the cloud more reliable. Several case studies can corroborate the issue about SLA automatic management (Amani et al. 2010, Brandic et al. 2010, Maurer et al. 2012, Emeakaroha et al. 2012a, Cedillo et al. 2014), concerning security, law, ethics.
In this paper we present a formal model for SLAs management in cloud forensic readiness. Such model includes a set of rules for input, output, and intermediary computations. This is followed by the design model of cloud forensic readiness system architecture. In the next Section we present some related work about SLAs formal representation and monitoring.

**Table 1:** Cloud Forensics Main Challenges

| CF Challenges | Elasticity | Multiple Locations | VM | Broad Network Access | Third Party Service | Cross – Providers | SLA |
|---|---|---|---|---|---|---|---|
| Reduced data access | X | X | X | | | | |
| Lack of physical control | X | X | X | | | | |
| Lack of standard | X | X | X | | | | |
| Multiple log formats | X | | X | | | | |
| No timestamps synchronization | | X | | X | | | |
| No routing information | | X | X | X | | | |
| Lack of investigation expertise | | X | | | | X | |
| Inappropriate legal measures | | X | | | X | X | X |
| Multi - tenancy | X | | | | | | X |
| Multiple jurisdictions | | X | | | | | X |

## 2. Related Work

Several researchers proposed both formal models and some monitoring tools for SLAs. In Czajkowski (et al. 2002) a protocol for negotiating SLAs among several actors is designed, which includes different types of SLAs. In Skene (et al. 2007) the SLAs are formalized using set theory, where the concepts of actors, events, parties, and actions requirements are modelled for the purpose of determining the SLAs' degree of monitor-ability. In Paschke (et al. 2008) a framework called Contract Log is presented, which consists of different types of rules

abstracting the contractual contents; they are derivation, reaction, integrity, and deontic rules. All these rules are included in a homogeneous syntax and knowledge base. In Unger (et al. 2008) the concepts of Parties, SLA parameters, and service level objectives are used to formalize SLAs for Services Oriented Architecture; the aim is to provide a manner for aggregating more SLAs in a single business process. The same concepts are used in Gosh (et al. 2012), but modelled only by tuples. Ishakian (et al. 2011) also uses tuples to address the issue of verifying efficient workload co-location of real-time workloads; he applies some transformation rules to compute the target SLA. Moreover, about thirty research projects were granted by the EU commission and their results are reported in (EU 2013). All these projects concern with automating different aspects of SLAs lifecycle. Another formal model utilizing set theory, predicates, and tuples is presented in De Marco (et al. 2014) with the purpose of comparing activities changing Cloud resources-status with some SLAs constraints. A framework called DESVI (Emeakaroha et al. 2010) is dedicated to monitor low-level cloud resources metrics, in order to detect whether their values respect the constraints extracted from some SLAs clauses, with the principal goal of detecting QoS violations. In Morshedlou (et al. 2014) SLA violations are considered from the user satisfaction point of view; thus, the main goal is to implement proactive resources allocation for reducing the negative impact of SLAs' violations and improving the level of users' satisfaction. In order to prevent clauses violations and the related cyber-crimes, Amani (et al. 2010) propose to add a phase dedicated to the evaluation of the contract itself to an SLA life cycle. Finally, some other researches are aimed to manage all the SLAs life cycle phases, namely the services requirements definition, the negotiation between providers and customers, the resources brokering, to the matchmaking and the monitoring of the agreed terms and conditions (Comerio et al. 2014, Wieder et al. 2011).

### 3. SLA Management in Cloud Forensic Readiness

A Cloud Forensic Readiness capability for SLA management (SLACFR) is aimed to record some information about the cloud behaviour with respect to SLAs. Such information is structured as a set of events, responsible of tracking the relationship between an attribute of a cloud entity and a (set of) constraint(s) on that attribute at time *t*. The input of such capability is composed of both information about some Cloud attributes and some SLA constraints, all represented with formal rules, as we describe in Section 4. The action flow of the capability depicted in Figure 1 begins on the availability of the contract(s) to monitor. The text is properly parsed via information extraction techniques (Grisham 1999) and transformed into a set of formal rules without repetitions. The cloud information is gathered from some services monitoring tools; they can represent network traffic information, resources statuses, file exchange operations tools, resources allocation, and measures facilities. The forensic readiness capability is responsible to record and output some events representing the matching between cloud attributes and the related SLA clauses. In particular, the capability recognizes in a real time manner whether some suspicious events happen: they represent a violation of a contractual constraint, called hypothesis. Such events recording represent some pre-investigative activities.

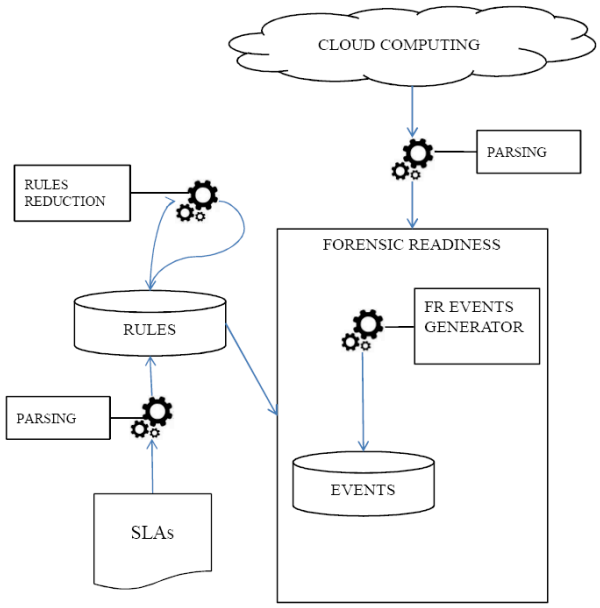


**Figure 1:** SLA Management in Cloud Forensic Readiness

## 4. Formal Model

The SLA Management Cloud Forensic Readiness (SLACFR) formal model is aimed to provide a theoretical approach to represent the management of SLAs contracts for cloud computing services in the context of forensic readiness. It is necessary to mention that a previous draft of the same formalism was discussed in De Marco (et al. 2014), where only a specific scenario has been taken into account. We extend the formal model by structuring the entities in a different manner and by adding some missing information, such as cloud architecture components. The formal model utilizes mathematical formalisms as tuples, set theory, functions (Ben-Ari 1993). Through these formalisms we provide a set of definitions about the entities manipulated by the architecture illustrated in Figures 1 and 3.

### 4.1 Cloud Computing

A cloud computing architecture (Mell et. al 2011) is object of monitoring activities by the forensic readiness capability discussed in Section3. Let *C* be the set of cloud architectures.

$$C = \{c_1, c_2, c_3, \dots, c_i\},\ i \in N$$

A cloud architecture *c* is composed of at least 2 data centres, then *c = D,* where *D* is the set of data centres.

$$D = \{d_1, d_2, d_3, \dots, d_i\},\ j \in N;\ |D| > 1$$

A data centre *d* is an entity containing one or more physical and virtual machines; the data centres are connected each other, composing networks. Let *d* be a data centre belonging to the set of data centres *D*, it is described by the tuple

$$d = (P, V, vmm, N^d); \qquad N^d \subseteq CN$$

Where *P* is the set of physical machines, *V* of virtual machines, *vmm* is a Virtual Machine Monitor, and $N^d$ is the set of data centre *d* is connected with. $N^d$ is subsect of the set of all the connections *CN*. Each element of the set $N^d$ describes the pairs of data centres connected each other.

$$CN = \{n_1, n_2, n_3, \dots, n_k\},\ k \in N; \qquad n = (d_1, d_2)$$

**Lemma**

In a cloud computing architecture there exist at least 2 data centres such that $|CN| \neq 0$

**Proof**

By definition $|D|>1$, then at least $D= \{d_1, d_2\}$, where $d_1 = (P, V, vmm, N^d{}_1)$ and $d_2 = (P, V, vmm, N^d{}_2)$

$N^d{}_1 = \{n_1\}$; $n_1 = (d_1, d_2)$

$N^d{}_2 = \{n_2\}$; $n_2 = (d_2, d_1)$

$N^d{}_1 \subseteq CN$ and $N^d{}_2 \subseteq CN \Rightarrow CN = N^d{}_1 \cup N^d{}_2 \Rightarrow CN = \{n_1, n_2\} \Rightarrow |CN| \neq 0$ .

Both physical and virtual machines are composed of a set of resources *R,* either software or hardware.

$$P = \{ R^p \mid R^p \subseteq R\};\ V = \{ R^v \mid R^v \subseteq R \};\ R = \{r_1, r_2, r_3, \dots, r_i\},\ j \in N$$

A resource *r* is described by a set of attributes *A* and their values, e.g., filename, date of creation, size. Let *a* be an attribute, it belongs to the set of attributes *A*.

$$A = \{a_1, a_2, a_3, \dots, a_j\},\ j \in N \qquad r = \{A^r \mid A^r \subseteq A\}$$

During the execution of a cloud service, the value of a resource attribute is subject to changes via operations. An operation is described by a mathematical tuple composed of a sender *s* triggering it, an operation result *value(r, a)*, an operation resource *r*, an attribute *a*, and an operation time $t_o$. The operation *value* is a mathematical function that changes the value of an attribute of a resource at the time *t-1* into another value at the time *t.* Such value is expressed with a unit measure *u*. A sender *s* is an entity performing operations in the cloud, either a human or a system process, e.g., an Internet session or a Dropbox process. Let *s* be a sender, it belongs to the set of senders *S*.

$$S = \{s_1, s_2, s_3, \dots, s_h\},\ h \in N$$
$$o = (s, a^r, value(a^r), u, t_o)$$
$$value: a^r_{t-1} \rightarrow value(a^r)_t$$

**4.2 Service Level Agreement**

As aforementioned, SLAs are contractual documents written in natural language composed of a set of information structured as clauses. In our formal representation, an sla *l* is an element of the set of slas *L*. An sla is described by a mathematical tuple composed of a set of hypothesis *H*, the validity starting time $t_{start}$ and the validity ending time $t_{end}$.

$$L = \{ l_1, l_2, l_3, \dots, l_i \},\ i \in N \qquad l = (H, t_{start}, t_{end})$$

A hypothesis *h* is an element of the set of the hypothesis *H*. A hypothesis is described by a mathematical tuple composed of the conditioned value *c k u* of the attribute $a^r$, the hypothesis validity starting time $t_s$ and the hypothesis validity finishing time $t_f$. A condition *c* belongs to the set of conditions *C*; a value *k* is related to the $a^r$ attribute; *u* is the unit measure used to express the value *k*, belonging to the set of unit measures *U;* the conditioned value *c k u* has to be verified during the time interval $t_f$ - $t_s$. In case no condition is expressed the default value is set as '='.

$$H = \{h_1, h_2, \dots, h_j\},\ j \in N \qquad c \in C \qquad u \in U$$
$$h = (c\ k\ u, a^r, t_s, t_f) \qquad C = \{\leq ; \geq ; < ; > ; <>; = \} \qquad U = \{u_1, u_2, \dots, u_l\},\ l \in N$$
$$t_f \geq t_s$$

Natural language is source of ambiguity for automatic recognizers. A manner to address this challenge is represented by the usage of the algorithm depicted in Figure 2; it needs the utilization of information extraction techniques (Grisham 1999), e.g., pattern recognition. The algorithm correctly identifies what are the words to either consider or discard. The computation recognizes a set of words representing the information needed by the formalism to populate specific sets, such as resources, attributes and unit measure, as discussed before, which contain elements to be considered without being combined with anyone else. Conversely, the set of hypotheses H includes information derived from the other three domains. The output is represented by the population of the sets of resources, attributes, unit measures, and hypotheses.

**Algorithm 1** Text to Formal Rules

```
1:  procedure T2FR(SLA)                                   ▷ Textual SLA
2:      Information Extraction techniques set up parameters;
3:      while Textual SLA not ended do
4:          current ← InformationExtractiontechniquesoutput;
5:          if currentisrecognized then
6:              PopulateR||A||U;
7:          else
8:              discardcurrent;
9:          end if
10:     end while
11:     H ← combinationofR,A,U;
12:     return R,A,U,H;                                   ▷ The sets are populated
13: end procedure
```

**Figure 2:** Text to Formal Rules Algorithm

**4.3 Forensic Readiness**

A cloud forensic readiness capability *f* is an element of the set of forensic readiness capabilities *F*. A capability is described by a mathematical tuple composed of a set of forensic readiness events *E* and a set of SLAs *L*.

$$F = \{ f_1, f_2, f_3, \dots, f_p \},\ p \in N \qquad f = (E, L) \qquad E \neq \emptyset \qquad L \neq \emptyset$$

An event *e* belongs to the set of events *E*. An events is described by a mathematical tuple composed of an operation *o*, a hypothesis to verify *h*, and the match time $t_e$.

$$E = \{ e_1, e_2, e_3, \dots, e_q \},\ q \in N \qquad e = (o, h, t_e) \qquad t_e \geq t_o$$

Given an event $e \in E$ and a hypothesis $h \in H$, the event is considered suspicious if the value of the operation $o$ on the attribute $a$ of the resource $r$ at the time $t$ is different from the conditioned value $c\ k\ u$ of the related hypothesis $h$ about the same attribute $a$ of the same resource $r$; the validity has to be determined during the correct time interval, namely the time of the operation assigning the value to the attribute $a$ of the resource $r$ $t_o$ has to be included in the time interval of the aforementioned hypothesis $t_f(h) - t_s(h)$.

$$value(a^r)_t \neq cku \qquad t_s(h) \leq t_o \leq t_f(h)$$

**4.4 Example**

The aim of this section is to illustrate with a conceptual example how an SLA violation is detected by the forensic readiness capability. The SLA clause suitable to help the reader to understand the SLACFR behaviour is derived from the public SLA for Amazon S3 (Amazon 2015):

*"Amazon Web Services will use commercially reasonable efforts to make Amazon S3 available with a Monthly Uptime Percentage [...] of at least 99.9% during any monthly billing cycle."*

We assume that dedicated information extraction techniques have been properly designed; they are required to populate the sets defined in Section 4.2 by correctly detecting the textual elements, and producing the following results.

**Resources.** The set of resources is populated as it follows: R = {AmazonS3}.
**Attributes.** The set of attributes is populated as**:** A = {available}.
**Unit Measures.** The unit measure *u* is not expressed, but we can correctly store the symbol *'%'* because it will be necessary during the computation of the match among operations and hypothesis, namely the events. The set of unit measures is populated as: U = {%}.
**Hypotheses.** The population of this set is a combination of the information already available, i.e., the elements stored in R and A. By scanning the text using as keywords the elements of R and A, we obtain that the conditioned value $c\ k\ u$ of the corresponding hypothesis is *'≥ 99%'*. The condition *c* of the hypothesis is *'≥'* because in the text the condition is expressed as *'at least'*. Finally, the starting and finishing times are derived by the text; in particular, both the starting and finishing time are calculated through the duration expressed in the text; because no particular hour constraint is written but the month is considered in a billing context, then the procedure assigns midnight as default.

$$H = \{h_1\}$$
$$h_1 = (\geq 99\ \%, available^{AmazonS3}, 01/01/2015\text{-}00{:}00, 01/02/2015\text{-}00{:}00)$$

**Senders, Operations, Events.** SLACFR has access to network channels and the Virtual Machines Manager of the underlying cloud platform. Such information is fed to the dedicated parsers (as depicted in Figure 1), which outputs the set of senders *S* and operations *O*. The element $IPa_1$ of the set *S* is a short form for $IPaddress_1$, which we assume is the IP address detected on the monitoring tool communicating with AmazonS3. A set of operations is then translated, and some events are generated by the engine FR Events Generator (see Figure 3).

$$S = \{IPa_1\}$$
$$O = \{o_1, o_2, o_3, o_4\}$$
$$o_1=(IPa_1, available^{AmazonS3}, 99\%, 02/01/2015\text{-}15{:}00)$$
$$o_2=(IPa_1, available^{AmazonS3}, 98\%, 05/01/2015\text{-}17{:}00)$$
$$o_3=(IPa_1, available^{AmazonS3}, 99\%, 10/01/2015\text{-}09{:}00)$$
$$o_4=(IPa_1, available^{AmazonS3}, 99\%, 05/02/2015\text{-}15{:}00)$$
$$E = \{e_1, e_2, e_3\}$$
$$e_1=(o_1, h_1, 02/03/2012\text{-}15{:}00)$$
$$e_2=(o_2, h_1, 05/01/2015\text{-}17{:}00)$$
$$e_3=(o_3, h_1, 10/01/2015\text{-}09{:}00)$$
$$e_4=(o_4, h_1, 05/02/2015\text{-}15{:}00)$$

The FR Events Generator component is also capable of properly labelling the events as suspicious or not, and to discard them depending on the time they are recorded at. In this case, the system will recognize event $e_2$ as suspicious because the conditioned value of the hypothesis is violated, indeed it is *98%* while it is supposed to be '*≥99%'*; similarly, event $e_4$ is discarded because recorded on a time outside the time interval established in $h_1$, namely after *01/02/2015-00:00*.

**5. SLACFR Architecture and Requirements**

In this section we describe an SLACFR capability model, by highlighting what are the inputs, processes, and outputs of the various components described in Figure 1. The diagram in Figure 3 uses the Business Process Model Notation (BPMN); it has been considered useful due to its capability of differentiating among independent functionalities, represented as business processes, each included in a swim-lane, i.e., the horizontal rectangles; the sub tasks necessary to accomplish a single purpose are located in the swim lanes. In the BPMN model for SLACFR we have three swim-lanes, where the topmost and the bottommost are responsible to compute the input necessary to the middle one, i.e., from the SLAs and the cloud infrastructure, respectively. Since a specific time, in the bottommost swim lane the monitoring tools output are acquired from a cloud platform and translated to obtain the formal rules about senders and operations, properly encrypted in order to respect some forensic best practices (CSA 2013, Grobler et al. 2013). The topmost swim lane, called Formalism Generation, is fed with the SLA documents. An initial engine implements the algorithm described in Figure 2 and transforms the contractual natural-language constraints to rules according to the formal model proposed in Section 4. Possible rules repetitions are reduced by the rules reduction sub-task. In the central swim lane, named Forensic Readiness, the detection of SLA-violation set of events is executed. The module called FR Events Generator is responsible of generating and storing the recorded events as soon as they happen, paying special attention to the suspicious events, responsible of identifying a contractual violation. It is not clear when such CFR business process has to terminate, because suspicious events can be recoded again and again. For the moment, we assume that the capability keeps running until the cloud organization considers it necessary for assuring more reliability to its services.

A system implementing such FR capability is dealing with computer security issues, and very likely managing private users' data; the main concerns are about the confidentiality of the information themselves. It is worth reminding that a CFRS will be deployed as an additional system, external to the Cloud infrastructure but capable of communicating with the necessary components to acquire the information from the Cloud (De Marco et. al 2013). Some dedicated, encrypted and secure communication channels are necessary at this point. Independently from the chosen technology, a reliable communication protocol must be adopted, allowing encrypting and decrypting the shared information. Moreover, the communication channels end-points might be secured by sophisticated mechanisms. The system must be reliable in terms of accessibility and data consistency; network connection falls cannot be admitted, as well as not updated or incomplete persistent storages. In terms of performance, the system has to be accessible 24h/7, due that crime prevention can happen in every moment of the day, every day; moreover, it should be required to manage big amount of data, and consequently be scalable from this point of view. Nevertheless, the network usage has to avoid channel saturation; indeed a proper medium connection is necessary, for instance optical fibre. The system has to be delivered and packaged in a multi-platform compatible manner, because it is not known a priori what the environment to interface with can be. Last but not least, it has to face with some investigative best practices. For instance, the investigation principles respected by the US National Institute of Justice (USJ 2011), by the UK ACPO (2007), the ones about data integrity (SWDGE 2006), and the Forensic Standard ISO IEC 27037 (CSA 2013) are widely adopted by the international community and considered necessary to be respected in case of dealing with potential forensic evidence.

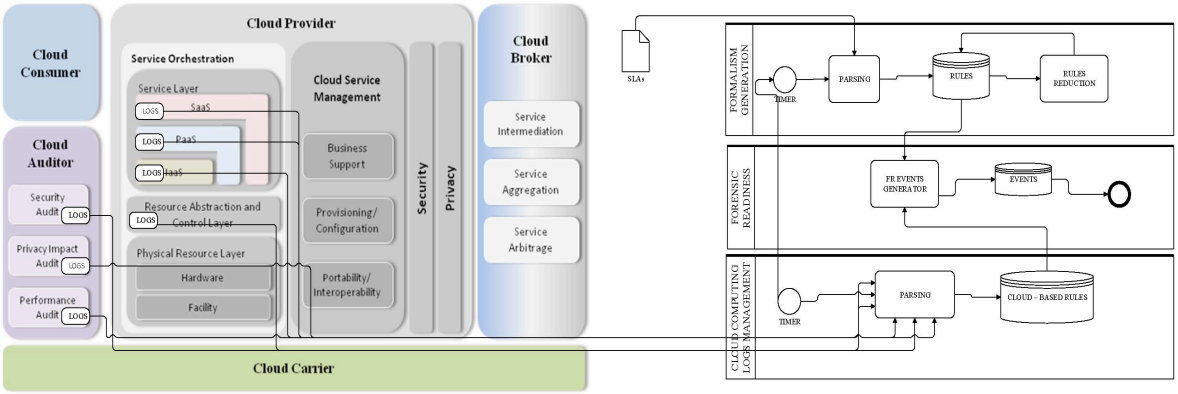


**Figure 3:** SLACFR BPMN Modelling

## 6. Discussion

SLACFR is devoted to implement a forensic readiness capability for SLA-related cloud crimes. The proposed approach concerns the representation of information from both the SLAs and the cloud in a specific formal model (see Section 4). The entities *sender, operation,* and *attribute* are necessary to reconstruct events timeline for forensic investigations. The resource attributes included in the *operation* entity have to match with the resource attributes expressed in the *hypothesis* to record forensic readiness events. The value of the operations on the attributes is compared with the conditioned value expressed in hypotheses concerning the same attributes. This match has to be performed by the FR Events Generator module of the architecture designed in Figure 3. Very likely, the actual source of cloud information can be represented by some cloud services logs. Such data can be gathered from dedicated logging tools included in the services furnished with a forensic readiness capability. If no logging tool is present, then an alternative information source needs to be identified. Is it worth to mention that log information is not structured in the same manner as expected by the formal model; then log information will be converted by a dedicated FR system component, to be added to the model in Figure 3. In order to calculate an operation resource attribute value, some metrics might be necessary. They compute the measure, necessary to the comparisons module, thus included in the system since its installation. Once the metrics for a resource attribute is applied and the value calculated, then the match with the value to be respected in the contractual constraint can be performed. In case the metrics are established by the contract itself, then such information has to be utilized, and the pre-existing metrics included in the system discarded. In this manner, the formal model demonstrates to be extensible on depending on the necessary entities to represent. Because it is at very high level abstraction, it can be enriched with additional information, such as some metrics or legal principles: specific rules can be added, and dedicated system modules interfaced with the existing ones presented in Figure 3.

## 7. Conclusion and Future Work

The management of SLA contracts in the context of cloud forensics is a very challenging research topic. Several proposals have been made in literature to guarantee QoS levels with the purpose of monitoring platforms behaviour for resources performances respect. In some cases, the issue is modelled and implemented; in other ones, specific frameworks are proposed to demonstrate how the monitoring is performed. In the future, we intend to prototype the SLACFR system for verifying some specific SLA constraints. The main goal is to demonstrate that SLACFR can really detect the exact time when a suspicious event is happening and recorded.
Moreover, we want to test the number of SLA violations detected in a specific time. We intend to simulate an attack to an IaaS Cloud service; this choice is led by the necessity of interacting directly with the physical resources level; moreover, PaaS and SaaS are built on top of IaaS.
The concern of the proposed FR capability prepares pro-actively the cloud for forensic, hence, a crime-reaction module is out of the SLACFR duties; nevertheless it can be considered as an extension or integration, capable of analysing a cloud crime scene with some information already available. We strongly believe that such forensic readiness capability can enhance the security strategies of a cloud platform, such that being considered as a must requirement for it, and very likely become a standard in the future.

**References**


ACPO Association of Chief Police Officers. (2007) *Good Practice Guide for Computer Based Electronic Evidence*, [online] http://www.acpo.police.uk/asp/policies/Data/ACPO\%20Guidelines\%20v18.pdf

Amazon Simple Storage Service (S3) (2015), [online] http://aws.amazon.com/s3/sla/ accessed on 22/02/2015

Amani, N., Hajipour, P., Seyedmostafaei, F. (2010) "An Appropriate Violation Detection Scenario for Service Level Agreements Based on WS-Agreement Protocol", *Journal of Convergence Information Technology,* Vol 5, No 1, pp. 40-47.

Baset, S.A. (2012) "Cloud SLAs: present and future", *ACM SIGOPS Operating Systems Review*, Vol 46, No 2, pp. 57-66.

Ben-Ari, M. (1993) *Mathematical Logic for Computer Science*, SPRINGER, first edition.

Birk, D. Wegener, C. (2011) "Technical Issues of Forensic Investigations in Cloud Computing Environments", *6th IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering,* pp. 1-10.

BPMN - Business Process Model and Notation, [online], http://www.bpmn.org/

Comerio, M., De Paoli, F., Palmonari, M., Panziera, L. (2014) "Web Service Contracts: Specification and Matchmaking", *Advanced Web Services, Springer, Athman Bouguettaya, Quan Z. Sheng, Florian Daniel Eds.,* pp.121-146.

CSA Cloud Security Alliance (2013) "Mapping the Forensic Standard ISO IEC 27037 to Cloud Computing", [online] https://cloudsecurityalliance.org/download/mapping-the-forensic-standard-isoiec-27037-to-cloud-computing/

Czajkowski, K., Foster, I., Kesselman, C., Sander, V., Tuecke, S. (2002) "SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed Systems" *Job scheduling strategies for parallel processing*, Springer Berlin Heidelberg, pp. 153-183.

De Marco, L., Kechadi, M-T., Ferrucci, F. (2013) "Cloud Forensic Readiness: Foundations", *Proc. ICDF2C Conference*, Springer International Publishing, LNICST series, vol. 132, pp. 237-244.

De Marco, L., Abdalla, S., Ferrucci, F., Kechadi, M-T. (2014) "Formalization of SLAs for Cloud Forensic Readiness", *Proc. ICCSM Conference*, Academic Conferences and Publishing International Limited, Reading, UK, Dr. Barbara Endicott-Popovsky University of Washington, Seattle, USA Edition, pp. 42 - 50.

Emeakaroha, V.C., Calheiros, R.N., Netto, M.A., Brandic, I., De Rose, C.A. (2010) "DeSVi: An Architecture for Detecting SLA Violations in Cloud Computing Infrastructures" *ICST Cloud Comp Conference*

European Commission (2013) *Cloud Computing Service Level Agreements - Exploitation of Research Results*, Directorate General Communications Networks, Content and Technology – Unit E2 – Software and Services, Cloud [online], Editor: Dimosthenis Kyriazis, http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?doc_id=2496

Ghosh, N., Ghosh, S.K. (2012) "An Approach to Identify and Monitor SLA Parameters for Storage-as-a-Service Cloud Delivery Model", *GC Wkshps*, pp. 724-729.

Grobler, C.P., Louwrens, C.P., von Solms, S.H. (2010) "A Multi-Component View of Digital Forensics", *Proc. ARES Conference*, pp. 647-652.

Grishman, R. (1997) "Information extraction: Techniques and challenges" *Information extraction a multidisciplinary approach to an emerging information technology*. Springer Berlin Heidelberg, 10-27.

Ishakian, V., Lapets, A., Bestavros, A., Kfoury, A. (2011) "Formal Verification of SLA Transformations" *IEEE World Congress on Services*, pp. 540-547.

Mell, P., Grance, T. (2011) *Final Version of NIST Cloud Computing Definition* [online], http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

Mishra, A.K., Matta, P., Pilli, E.S., Joshi, R.C. (2012) "Cloud Forensics: State-of-the-Art and Research Challenges", *International Symposium on Cloud and Services Computing*, pp. 164-170.

Morshedlou, H., Meybodi, M.R. (2014) “Decreasing Impact of SLA Violations: A Proactive Resource Allocation Approach for Cloud Computing Environments”, *IEEE Transactions on Cloud Computing*, Vol 2, No 2, pp. 156-167.

NIST National Institute of Satandards (2014) “Cloud Computing Forensic Science Challenges” *NIST Cloud Computing Forensic Science Working Group Information Technology Laboratory*, [online], http://csrc.nist.gov/publications/drafts/nistir-8006/draft_nistir_8006.pdf

Palmer, G. (2001) “A Road Map for Digital Forensic Research Technical Report”, *Technical Report from DFRW Workshop,* Utica, New York.

Paschke, A., Bichler, M. (2008) “Knowledge Representation Concepts for Automated SLA Management”, *Decision Support Systems,* Vol 46, Issue 1, pp. 187-205.

Patel, P., Ranabahu, A.H., Sheth, A.P. (2009) “Service Level Agreement in Cloud Computing” [online] http://corescholar.libraries.wright.edu/knoesis/78

Reilly, D., Wren, C., Berry, T. (2010) “Cloud Computing: Forensic Challenges for Law Enforcement”, *Proc. ICITST Conference*, pp. 1-7.

Reilly, D., Wren, C., Berry, T. (2011) “Cloud Computing: Pros and Cons for Computer Forensic Investigations”, *International Journal Multimedia and Image Processing*, Vol 1, No 1, pp. 26-34.

Rowlingson, R. (2004) “A Ten Step Process for Forensic Readiness”, *International Journal of Digital Evidence*, Vol 2, No. 3, pp. 1–28.

Ruan, K., Carthy, J., Kechadi, M-T., Crosbie, M. (2011) “Cloud Forensics: an Overview”, *Proc. IFIP Conference*, Vol 7.

Ruan, K., Carthy, J., Kechadi, M-T., Baggili, I. (2013) “Cloud Forensics Definitions and Critical Criteria for Cloud Forensic Capability: An Overview of Survey Results”, *Digital Investigation*, Vol 10, No 1, pp. 34-43.

SWDGE Scientific Working Group on Digital Evidence (2006) *Data Integrity Within Computer Forensics*, [online] https://www.swgde.org/documents/Current\%20Documents/2006-04-12\%20SWGDE\%20Data\%20Integrity\%20Within\%20Computer\%20Forensics\%20v1.0

Skene, J., Skene, A., Crampton, J., Emmerich, W. (2007) “The Monitorability of Service-Level Agreements for Application-Service Provision” *Proc. International Workshop on Software and Performance*, pp. 3-14.

Tan, J. (2001) “Forensic Readiness, Technical Report”, *@Stake Organization*, Cambridge, MA, USA, [online] http://isis.poly.edu/kulesh/forensics/forensicguide_readiness.pdf

Unger, T., Leymann, F., Mauchart, S., Scheibler, T. (2008) “Aggregation of Service Level Agreements in the Context of Business Processes”, *Proc. ICEDOC Conference*, pp. 43-52.

USJ U.S. National Institute of Justice (2011) *Electronic Crime Scene Investigation: A Guide for First Responders* [online] http://www.ncjrs.org/pdffiles1/nij/187736.pdf

Wieder, P., Butler, J. M., Theilmann, W., & Yahyapour, R. (2011) “Service level agreements for cloud computing”, Springer.