Dynamic Security Assurance in Multi-Cloud DevOps

Erkuden Rios[1], Eider Iturbe[1], Wissam Mallouli[2], Massimiliano Rak[3]
[1] ICT Division, Fundación Tecnalia Research & Innovation. Derio, Spain 48160
Email: {erkuden.rios, eider.iturbe}@tecnalia.com
[2] Montimage Research & Development. Paris, France 75013
Email: wissam.mallouli@montimage.com
[3] Univ. of Campania Studies Luigi Vanvitelli, Naples, Italy 81031
Email: massimiliano.rak@unicampania.it

***Abstract*—Development and operation of multi-cloud applications, i.e. applications which consume and orchestrate services from multiple independent Cloud Service Providers, are challenging topics nowadays. Systematically addressing security assurance in such applications is an additional issue, unsolved at state of art. This paper introduces the MUSA DevOps approach to holistic security assurance in multi-cloud applications and details particularly the proposed approach to dynamic assurance at operation phase, which enables to early feed back the application security status to the development phase in order to take corrective actions as soon as possible, whenever they are needed.**

I. Introduction

The rise of Cloud Computing advocates for a myriad of cloud providers and cloud services that offer on demand computing power and resources at lower cost [1]. Hybrid and multi-cloud strategies that combine the use of multiple and heterogeneous services are taking off slowly, as security remains one of the major inhibitors of Cloud adoption [2].

Non-security expert developers need mechanisms and tools that help them in the creation of secure multi-cloud applications that dynamically can adapt to changes in both the cloud service composition and the context. For an early reaction to changing conditions and threats in the environment, easy to use and understand tools for monitoring and controlling security at operation are also needed.

This paper presents a solution to dynamic DevOps security assurance for multi-cloud applications that have their components deployed in distributed and heterogeneous clouds. The solution relies on the adoption of the MUSA approach [3] that allows for the agile development and operation of such applications considering security as a design tenet and driver for application development (e.g. in cloud service selection). This means that it enables to easily specify, deploy and asses at runtime the required security controls in distributed application components, following a DevOps approach. The solution is part of the MUSA framework which is the result of the EU H2020 research project named MUSA [4].

The MUSA framework comes in form of a single solution that seamlessly integrates a number of mechanisms supporting different steps in the (multi-)cloud-based application lifecycle: application modelling, risk analysis, cloud service selection based on security controls they offer, automatic generation of composite Service Level Agreement (SLA), multi-cloud deployment, and continuous assurance (monitoring and enforcement of security behaviour) to minimize risks at runtime.

The paper is structured as follows. After the introductory section, Section 2 clarifies the concept of multi-cloud and multi-cloud application, and describes the state of the art of security solutions for this type of environments. Then, Section 3 focuses on the challenges and related work to security assurance in multi-cloud. In Section 4, we introduce the complete MUSA workflow and framework for the security-intelligent lifecycle management of multi-cloud applications. In Section 5 we detail the proposed approach to Dynamic Security Assurance in multi-cloud DevOps, which is part of the MUSA framework. Section 6 discusses the benefits brought by the solution in two success stories in the domains of flight scheduling systems and smart mobility services. Finally, Section 7 concludes the paper and describes the future work.

II. Security in Multi-cloud environments and applications

The term *multi-cloud* is used in many different contexts and refers to the idea of accessing resources from multiple Cloud Service Providers (CSPs). As such, *multi-cloud computing* can be considered as a special case of *inter-cloud computing*, which has been defined in [5] as: *A cloud model that, for the purpose of guaranteeing service quality, such as the performance and availability of each service, allows on-demand reassignment of resources and transfer of workload through a interworking of cloud systems of different cloud providers based on coordination of each consumers requirements for service quality with each providers SLA and use of standard interfaces.*

Even if in the literature the terminology is not yet stable, [6] proposes to adopt the term *inter-cloud* as the generic term indicating the adoption of multiple CSPs. The term *cloud federation* denotes a set of CSPs that voluntarily collaborate

978-1-5386-0683-4/17/$31.00 ©2017 IEEE

(e.g. interconnect their infrastructures) to allow sharing of cloud resources among themselves to serve each other or their consumers, while the term *multi-cloud* refers to the usage of cloud resources from multiple CSPs without the need to exist a previous explicit collaboration agreement or interconnection among the service providers.

In this sense, we can define a *multi-cloud application* as an application which their components use or are deployed in cloud resources from multiple independent CSPs, and therefore, may use or be distributed over heterogeneous CSPs.

At state of art, the multi-cloud topic is considered extremely relevant: the need for solutions addressing multi-cloud environments is well demonstrated by the number of European research projects that are proposing solutions and techniques to address the multi-cloud approach, like OPTIMIS [7] , mOSAIC [8] , MODAClouds [9] , PaaSage [10] , Cloud4SOA [11]. It is out of the scope of this paper to offer a complete survey of such activities, we suggest the interested reader the following papers: [5] [6] [12].

Security in multi-cloud environments is an open topic and analysed in the literature from two divergent points of view. Part of the existing literature outlines that multi-cloud solutions enable system security improvement, while other authors, on the contrary, believe that the multi-cloud paradigm brings new security risks and vulnerabilities. The authors of [7] [13] offer simple surveys of solutions that try to improve the security using multi-cloud techniques, usually focusing on cloud storage services. We suggest the interested reader to focus on the above survey and the paper they refer. On the other hand, [13] [14] face the security in multi-cloud applications from a different perspective: they analyse different multi-cloud solutions and try to make a security assessment of the overall application behaviour. According to such vision, multi-cloud is open to new security threats that decreases the global security level.

III. Security Assurance in multi-cloud

Looking closer to cloud security assurance, even if not particularly oriented to multi-cloud, there exist a number of Cloud systems monitoring solutions such as those collected in the surveys provided in [15] and [16].

The challenges for cloud monitoring identified in such works remain the same in multi-cloud setups but complexity highly scales due to multiplicity of cloud services and types of compositions. The MOSAIC approach [8] and PaaSage approach [17] were two of the initiators of the multi-cloud monitoring problem research.

The state of the art multi-cloud monitoring solutions mainly focus on elasticity policies and quality of service (QoS) but lack specific support to security control. This is the case of the framework presented in [18] which relies in the enactment of the application model (written in CloudML language [19]). The solution for model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems, was developed as part of the REMICS, ModaClouds and PaaSage projects. Further, monitoring is dependent on the definition of the metrics in CloudML language in the application model, rather than in an standard Service Level Agreement (SLA) format (such as Web Service Agreement), which limits the approach.

Similarly, the cross-layer monitoring framework described in [20] is based on event patterns that allow for triggering adaptation rules.

The SeaClouds project [21] aims at adaptive management of complex applications deployed across multiple clouds by supporting the distribution, monitoring and migration of application modules over multiple heterogeneous PaaS. Again, the focus is on assuring the QoS of the complex application but does not address specifically the security issues.

Initiating the path towards security assurance, the CUMULUS project [22] delivered an integrated framework of models, processes and tools supporting the certification of security properties of cloud services (IaaS, PaaS or SaaS).

On security specifics, the SPECS project [23] delivered an open source framework to offer Security-as-a-Service, by monitoring security parameters specified in SLAs, and also providing the techniques to systematically manage SLAs lifecycle. The project provided solutions for automatic negotiation and monitoring of SLAs between CSPs and SPECS platform based on security properties of cloud services. The work presented in this paper directly links with the outcomes of SPECS as MUSA extends these to multi-cloud setups.

At the best of authors knowledge, no previous work addresses the dynamic multi-cloud security assurance in a holistic and systematic way, i.e. no previous approach combines security-by-design principles to multi-cloud application, identifying the security risks that multi-cloud applications are exposed to and dynamically orchestrating security mechanisms to mitigate the risks outlined by the security analysis at runtime, thus directly aligning the (re-)design decisions with security properties assessment at runtime.

IV. The MUSA DevOps framework for multi-cloud applications

Our approach to dynamic assurance in multi-cloud relies in holistic support to security, in the sense that security is addressed in all the phases of the multi-cloud application lifecycle, and by a multi-disciplinary team that combines expertise from diverse aspects of cloud system engineering and management.

Therefore, MUSA promotes the DevOps paradigm [24] since an integrated team of joint forces from Development and Operations teams collaborates in both the multi-cloud application engineering and service offering processes. We name *DevOps Team* to such multi-disciplinary team that involves application architects, developers, security architects, business managers, service operators and system administrators. Note that in each of the lifecycle phases, from design to service operation, one of the roles in the team may act as the prominent actor with the responsibility of the final decision. Still, the approach promotes the collaboration of the group for the objective of early understanding the causes of the security incidents and prompt reaction to them.

The MUSA DevOps approach is enabled by the MUSA framework which combines in a single workflow predictive and reactive security mechanisms, as shown in Fig.1.

The workflow supports dynamic assurance in DevOps which consists in the combination of the following activities in an iterative process:

1) *Modelling of the application cloud and security requirements*: The first step in the multi-cloud application design is the specification of the Cloud Provider Independent Model (CPIM) of the application, a task supported by the MUSA Modeller. The CPIM, captured in a MUSA extended CAMEL language, is the specification of the multi-cloud application in a level of abstraction independent from specific Cloud services and providers the application will use.
2) *Continuous Risk Assessment* that helps in the selection of the security controls and metrics that will be granted in the Security SLA and controlled at runtime. The activity follows a methodology similar to the one described in [25]. It allows for selecting the relevant threats according to the component nature, evaluating the technical and business impact of the threat exploitation, as well as, in order to minimize such impact, defining the desired countermeasures or controls required over the cloud services the components will use or be deployed in. The risk assessment is continuously updated with the feedback from the continuous monitoring of the controls behaviour at runtime.
3) *The Cloud services selection* relies on the use of the MUSA Decision Support Tool (DST). In order to take most out of cloud services combination in terms of security, the DevOps Team is supported in the selection of cloud services that best match the security requirements of the multi-cloud application components. The best match is calculated by comparing the security controls offered by the cloud services under study (those previously categorized in the MUSA CSP Data Repository) with the security requirements of the individual components.
4) *Multi-cloud application components security SLA templates generation*. Once the most appropriate cloud service is selected for each of the components, the DevOps Team will use the MUSA SLA Generator to automatically create the Security SLA templates of the components. The Security SLA templates define the required security Service Level Objectives (SLOs) of the components in the basis of the SLOs required over the cloud services that they will use. For this, the MUSA framework supports the verification of the feasibility of the components Security SLA templates by checking whether the cloud service offerings selected in the previous step do offer such security requirements (in form of security controls). In case they do not, the MUSA security enforcement agents may be adopted to offer them.
5) *Multi-cloud deployment planning*. The MUSA Deployer will generate the deployment Implementation plan for the multi-cloud application. The Implementation plan specifies the applications software components to be installed, and the cloud services to be provisioned, as well as their configuration details, in order to run the components satisfying the security requirements.
6) *Multi-cloud application Composite Security SLA generation*. In this step the DevOps Team is supported in the automatic generation of the final offered Security SLA of the multi-cloud application. The Security SLA of the overall application is the result of the composition of the individual components Security SLAs, i.e. it considers the Security SLAs of individual components as well as the component nature (e.g. web server, database, etc.) and the relationships between the components (e.g. uses, is deployed in, protects, etc.). The last step in design process will therefore be the Security SLA composition activity.
7) *Multi-cloud deployment execution*. The DevOps Team uses the MUSA Deployer to execute the Implementation plan, i.e. to provision and configure the needed cloud resources as well as deploy both the multi-cloud application components and the corresponding MUSA agents required in the plan.
8) *Continuous Monitoring of the application Security SLA* once the components are deployed and running, and early feedback to development. Finally, at runtime or operation phase, the MUSA Security Assurance platform starts monitoring the multi-cloud application based on the final SLAs and the Implementation plan. In case potential or actual violations of the SLA are found reaction measures such as remodelling the application or re-evaluating risks again are recommended.
9) *Dynamic adaptation of the multi-cloud application* to meet the security status guaranteed in the Security SLA. The MUSA Security Assurance platform also supports the dynamic enforcement of secure behaviour of the application by means of activation of MUSA security enforcement agents. The agents are activated as a reaction mechanism to a security problem detected in previuos step. The monitoring step is informed on the status of the activation of the enforcement agents as well as on the required enforcement events.

Note that Modelling, Continuous Risk assessment and Cloud services selection follow an iterative loop that allows identifying whether there are any application security requirements that are not possible to be addressed with the security controls offered by the cloud services available (previously categorized). In this case, the DevOps Team should revisit the CPIM to include protection components or specify the use of MUSA security enforcement agents that offer such missing security controls (if available).

The two last steps are the core of the Dynamic assurance in multi-cloud and will be further detailed in next section.

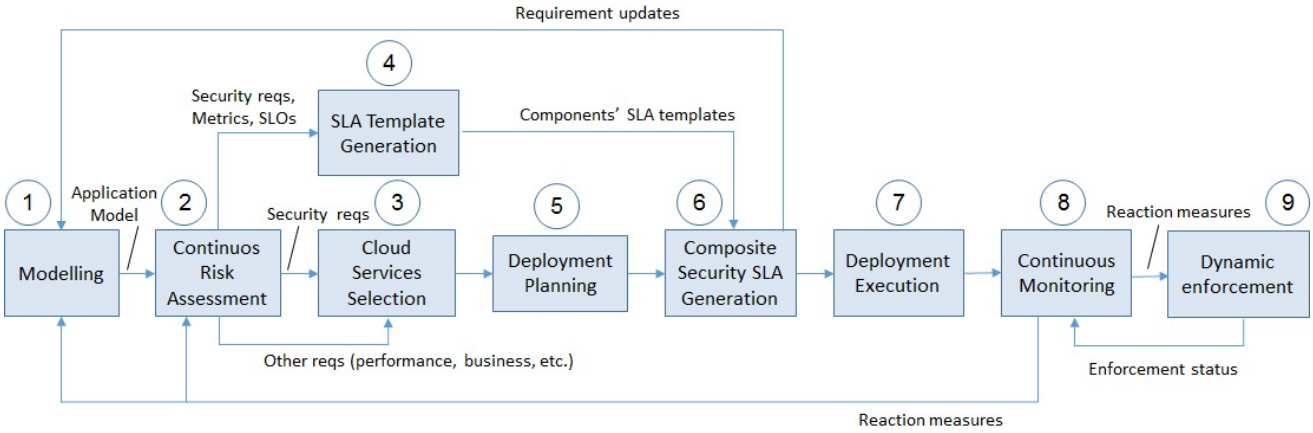

Fig. 1. Overall security DevOps workflow supported by MUSA framework.

V. MUSA Security Assurance in Multi-cloud DevOps

The runtime operation of the multi-cloud application starts once all the components of the application have been appropriately deployed and the application is running. Depending on the architecture of the application, some of the components may be deployed in cloud infrastructures or may use PaaS or SaaS services. Therefore, the runtime environment scenarios may be diverse and the proposed solution in MUSA needs to be instantiated to select the right monitoring and security enforcement agent that fulfill the selected environments needs.

A. MUSA Security Assurance Workflow

As shown in Fig.2, the MUSA runtime support consists of two main activities, explained below.

1) Continuous Monitoring of Security SLAs fulfilment: The objective of this activity is to monitor the runtime security behaviour of the selected multi-cloud application components in order to early react to possible security incidents. The activity involves:

- *Extraction of metrics and thresholds from Security SLAs*: After retrieving the offered composite Security SLA registered in the MUSA Security Assurance Platform, the platform will extract from it the security metrics that need to be monitored in the application components and in the cloud providers. The SLO thresholds that will apply for triggering the alerts and notifications are also learned from the Security SLAs.
- *Configuration of monitoring agents*: Make the needed arrangements and configurations for the MUSA monitoring agents to properly work and enable them to monitor the security metrics.
- *Security metrics measurement and monitoring*: Take the actual values of the metrics and store them in the Measurement Repository.
- *Reporting and visualization of monitoring results*: Show to the user the resulting values measured and the reports from the computation of the metrics.
- *Notification of security incidents*: The DevOps Team can subscribe to desired alerts and notifications. The envisaged notifications could be mainly of two types: Security SLA violations (when it is detected that a SLOs in the Security SLA is not reached) and alerts (when it is detected that a threshold level in the SLO is not reached, i.e. before any violation in the SLO occurs). The user will therefore need to set the threshold levels for the alerts.

2) Dynamic adaptation and reaction to security incidents: The goal of this activity is to decide and execute the needed reaction measures in case security incidents or Security SLA violations occur. The reaction to security incidents in MUSA relies on different mechanisms depending on the cause of the incident and whether it is an alert or a violation. In general, is up to the DevOps Team the decision of whether to react at the level of alert before any violation takes place. In the following, we provide a summary of the possible reaction measures involved in the process.

- *Activate security enforcement agent*: In those cases that the multi-cloud application component was prepared in the Design or Deployment time with a MUSA enforcement agent for enforcing a particular security control, it is possible to activate the agent at runtime if the detected security incident corresponds to that security control. The enforcement service in the MUSA Security Assurance Platform will be the one in charge of identifying and activating the needed MUSA enforcement agent in the component.
- *Re-deployment of multi-cloud application*: In case the cause of the security incident resides in the bad security efficiency of a selected cloud service, the DevOps Team may need to replace it with some other cloud service that can provide similar functionality and security properties. This means that the DevOps Team will need to look for a new Cloud Service combination for the multi-cloud application components and re-deploy the multi-cloud application, or at least the component that used the failing cloud service. Most likely the rest of cloud services in the

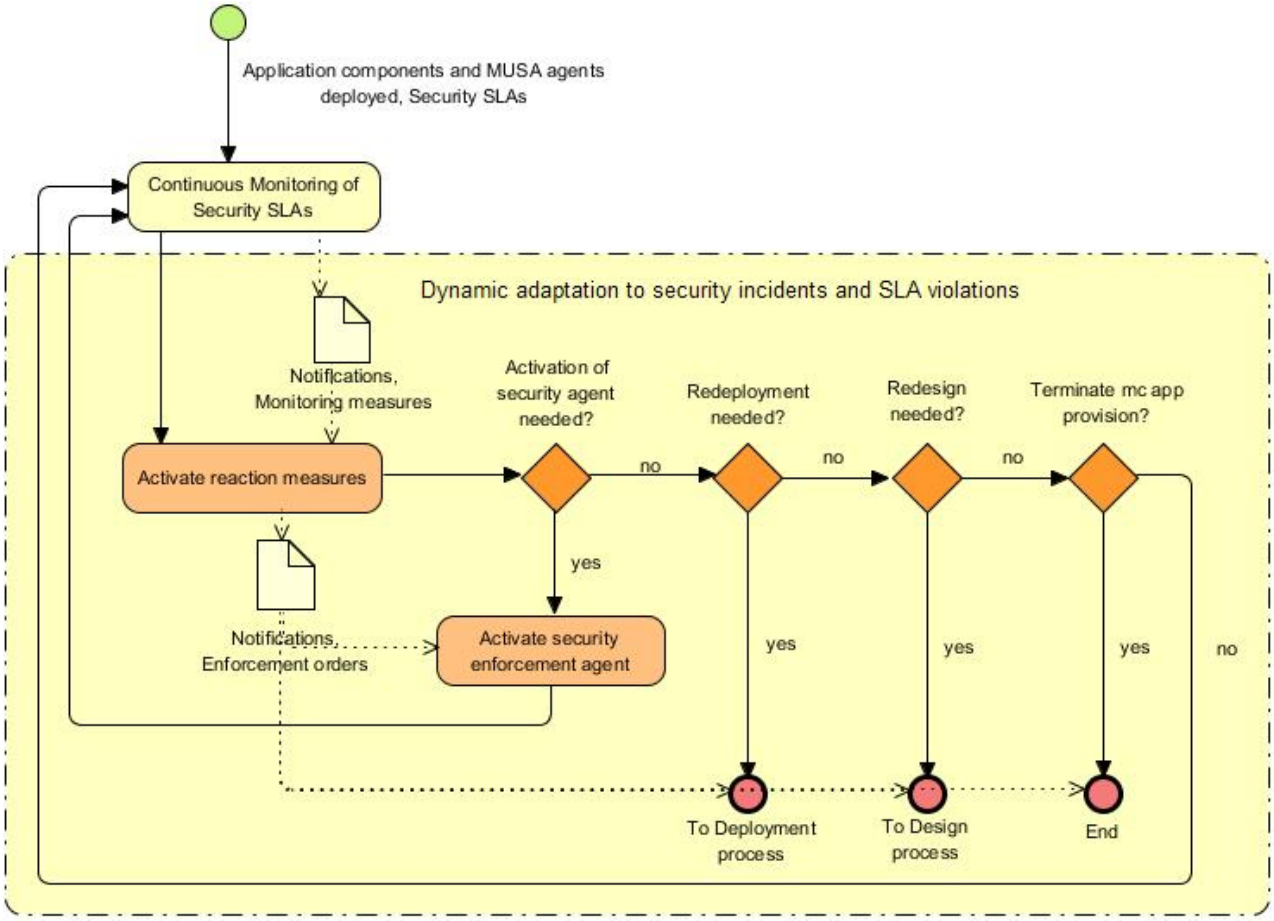

Fig. 2. MUSA Security Assurance Platform supported activities in multi-cloud application operation.

Cloud Service combination will not change, but in any case, the whole process should start from the beginning in order to make sure the new Cloud Service combination with the Cloud Service replacement still holds the multi-cloud application security requirements.

- *Re-design of multi-cloud application*: In case the cause of the security incident resides in an incorrect or poor security performance of a multi-cloud component and not in the Cloud Services in use, the DevOps Team may need to update the multi-cloud application design and refine the security requirements or modify the components. This means that the DevOps Team will need to analyse the report of the causes and start the Design process again.

B. MUSA Security Assurance architecture

The MUSA Security Assurance solution fits the operation phase of the MUSA framework and requires two main inputs to work properly:

- The Security SLA of the application to monitor: The MUSA Security Assurance platform retrieves the multi-cloud composite application SLA as well as the individual components' SLAs referred by it. From individual SLAs, the MUSA Security Assurance platform can monitor the security of single components, and from composite SLAs it can check the end-to-end security of the multi-cloud application taking the communication exchanges between remote components into account.
- The application deployment Implementation plan: From this plan, the MUSA Security Assurance platform recuperates the list of monitoring agents deployed with each application component as well as their IP addresses. This information is vital to link the monitoring agent with the application component to monitor the right security metrics that are specified in the application component security SLA.

The MUSA security assurance is composed, as depicted in Fig.3, of three main elements:

- The *MUSA Monitoring agents* responsible for collecting different security metrics and relevant events to be analyzed by the centralized MUSA Security Assurance platform (deployed as a service).
- The *MUSA Security enforcement agents* that are deployed and/or activated in case of any security incident detection.
- The *MUSA Security Assurance Platform* that allows collecting all the security metrics and events from individual application components (on the KAFKA event

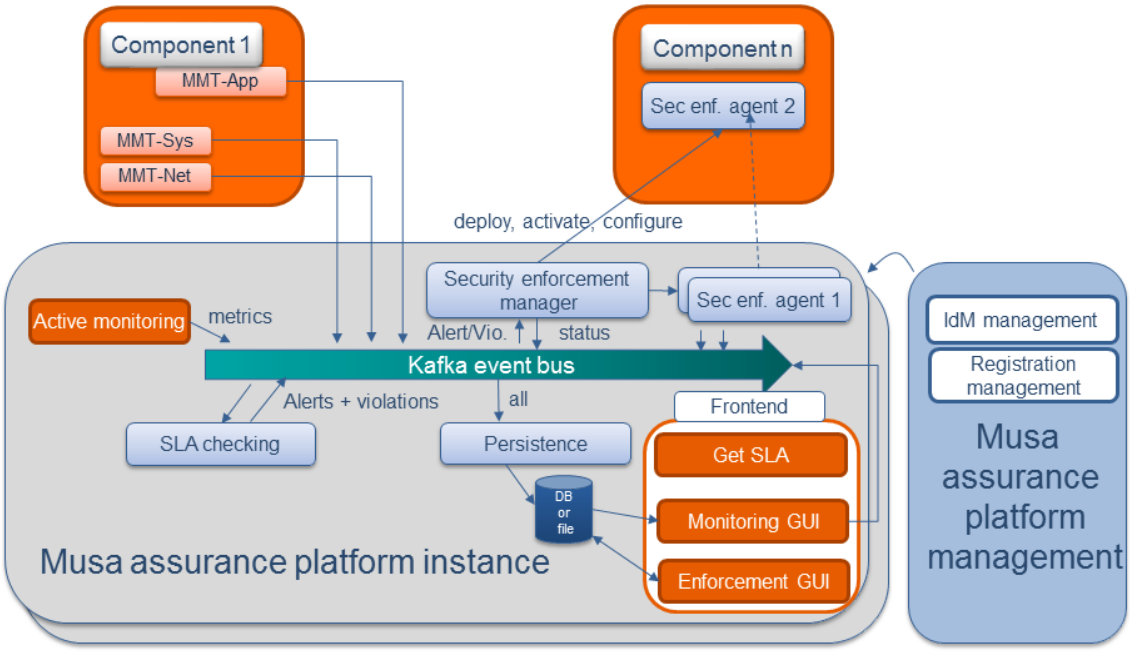

Fig. 3. MUSA Security Assurance Architecture.

bus), check the component Security SLAs and compute the composite metrics to check the global application Security SLA ("SLA checking" module). In case of an alert or a violation, the "security enforcement manager" is responsible for deploying, activating and configuring the local or remote security enforcement agents to mitigate the security risk. The default communication with these agents is the same KAFKA bus.

More details about the monitoring and enforcement agents are presented in the following subsection.

C. MUSA Security agents

1) MUSA Monitoring agents: The security metrics that can be gathered in the context of the MUSA framework are part of the security metric catalogue presented in [26]. Different monitoring agents are thus deployed in the same virtual machine or container as the application component to compute security metrics by relying on different sources: Network, operating system or application.

- *Network monitoring agent*: This type of monitoring agents is responsible of analyzing network traffic from different network interfaces of the virtual machine or container where the application components is running. It is composed of the following features:
  - Packets capture, filtering and storage
  - Security events extraction and statistics collection, and
  - Traffic analysis and reporting providing, network, application, flow and user-level visibility.
  - Security incidents detection.

  This agent facilitates network performance monitoring and operation troubleshooting through its real-time and historical data gathering. With its advanced rules engine, the monitoring agent can correlate network events to detect performance, operational and security incidents.
- *System monitoring agent*: Monitors operating system resources which may be the cause of server performance degradation, and spots performance bottlenecks early on. The agent relies on Linux top command, which is frequently used by many system administrators to monitor Linux performance, being available in many Linux/Unix-like operating systems. The top command is used to display all the running and active real-time processes in an ordered list updating it regularly. It displays CPU usage, Memory usage, Swap Memory, Cache Size, Buffer Size, Process PID, User, among others.
- *Application monitoring agent*: Monitors information about the internal state of the target system, i.e., multi-cloud application component to the MUSA Security Assurance platform during its operation. It notifies the MUSA Security Assurance platform about measurements of execution details and other internal conditions of the application component. The application monitoring agent is a Java library composed by two parts. The first is an aspect to be weaved into the application code via pointcuts in order to send application-internal tracing information to the MUSA Security Assurance platform for analysis. It is composed of a set of functions that can be weaved in strategic application points to capture relevant internal data. The second part connects the aspect with the notification tool via a connector library, providing a simple interface for sending log data to the MUSA Security Assurance Platform in a secure way. In other

words, the application monitoring agent is responsible for extracting the information from the system, and the connector is in charge of transferring it.

2) MUSA Security enforcement agents: The security enforcement services offered in MUSA are security controls or mechanisms that could be easily integrated in multi-cloud application components and activated at runtime whenever needed. The enforcement mechanisms were proposed to be built on top of existing open source solutions and the major innovation resides in having MUSA framework as single point of management for orchestrating multiple mechanisms that address diverse security properties on the multi-cloud applications. In the following, we present two main enforcement agents that are available in the framework for dynamically adapt the security behaviour of the applications.

- *The high availability (HA) framework*: The HA framework is a collection of open-source software built around the Corosync/Pacemaker stack [27], patched and configured to work together to bring clustering mechanisms to multi-cloud-based services. It provides the following functions: Automatic routing between services, inter-component communication security enhancement, load balancing, high availability, scaling, automatic failover and access control lists. The application component is assumed to be placed inside the Docker container, and the node itself is assumed to be IaaS. Other configurations are possible, such as side-by-side deployment of the framework and the service for both IaaS and PaaS systems. The framework itself can be deployed using MUSA Deployer just as a regular enforcement agent. Most of the cookbooks and packages are provided as open-source and can be found on Chef Supermarket.
- *The access control (AC) framework*: In multi-cloud environments where the application components are distributed over heterogeneous cloud providers there is the need to ensure that only authorised parties can access and use the functionality (services) offered by the components. The Access control framework designed and implemented in MUSA is an enforcement mechanism conceived to provide two major features: Access control in end-user-to-component communication, and access control in component-to-component communication. The framework uses solutions external to MUSA to offer the authentication and authorisation functionality and it does assume that the management of the multi-cloud application users is not done by MUSA but the multi-cloud application itself. The AC framework supports the control of only authorised components are granted access to services in other components. To this aim, the MUSA AC agent in the components should be accompanied by a Decision Agent that is able to evaluate the permission according to the access control policies pre-defined. This way, following the XACML [28] policy architectural model, the Policy Enforcement Point (PEP) and Policy Decision Point (PDP) would be included in each component, which allows taking the permission decisions locally and increase the performance.

VI. Validation in case studies

The initial proof of concept of the MUSA dynamic assurance approach has been evaluated in two real-world multi-cloud oriented applications:

- Flight scheduling application by Lufthansa Systems, Germany. This is a working prototype for a flight schedule planning application, aimed at being used by tens of airlines around the world. The prototype is realized as a multi-layered, distributed web application and provides a scalable platform of self-contained and loosely coupled business components, each capable of running in a separate process and interacting by use of lightweight REST style communication protocols. The assurance focus in this case study includes data integrity, confidentiality, localization and access control.
- Smart mobility service by Tampere University of Technology, Finland. This is an open data based multi-cloud application optimizes urban travel experience in Tampere city. MUSA assurance will facilitate the control of the needed privacy and protection for citizens mobility data.

The DevOps Team in each of the case studies followed the different steps described in section IV to model the composite application and assess the risk level for each of its components. Individual SLAs templates were generated and Cloud service providers selected using the DST tool. After the deployment planning of the application components and the computation of the composite application SLA, the application was deployed successfully using the MUSA Deployer tool.

The different monitoring agents described in section V.B.1 were deployed to retrieve the security metrics specified for the different application components. For instance, for the flight scheduling application, one of the application components is called ”fleet module”, responsible for providing the air fleet related services. After the risk analysis step, the SLA specified the following security controls (partial list), expressed according to the NIST control framework [29]:

- Information input validation (NIST SI-10)
- Penetration testing (NIST CA-8)
- Vulnerability scanning (NIST RA-5)
- Least privilege (NIST AC-6)
- Process isolation (NIST SC-39)
- Denial of service protection (NIST SC-5)
- Separation of duties (NIST AC-5)
- ...

And the following security metrics (partial list) as shown in Fig.4:

- Resilience to attacks
- M1-Level of redundancy
- M13- Scanning frequency
- Risk Assessment Vulnerability Measure
- Remote Access Control Measure
- ...

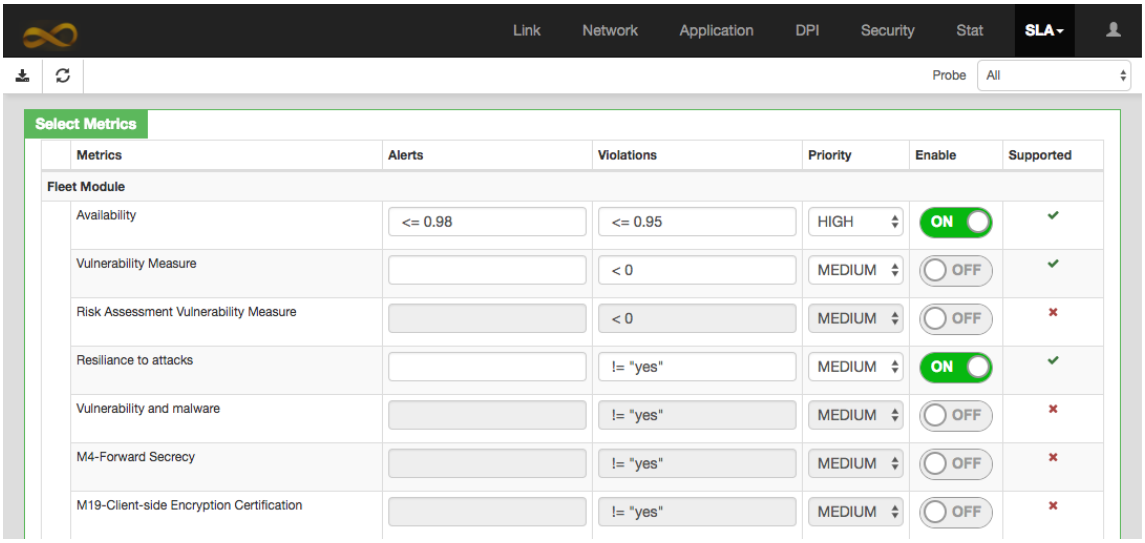

Fig. 4. Partial list of security metrics for the fleet module.

The continuous monitoring of the application allowed detecting potential malicious activities based on a set of detection rules denoting several kinds of attack signatures. To evaluate the efficiency of the solution, we emulated an unauthorized access to the flight scheduling application and the generation of badly formed internal messages on the events bus used by this application. This kind of activity is successfully detected by the monitoring agent and notified to the MUSA Security Assurance platform that raised immediately a violation alarm to the DevOps Team. A recommendation to deploy a stronger access control mechanism was also made. The DevOps Team followed the recommendation and the activation of the MUSA access control framework was performed and the access checking using the MUSA framework allowed to minimize the risk of such attack.

VII. Conclusion

Multi-cloud applications have to deal with the security of the individual components as well as with the overall application security including the communications and the data flow between the components. Despite the cloud service providers offer their own security controls, the multi-cloud application has to ensure integrated security across the whole composition. Therefore, the overall security depends on the security properties of the application components, which in turn depend on the security properties offered by the cloud resources they exploit.

In this context, the MUSA framework has been conceived and implemented to support the security-intelligent lifecycle management of multi-cloud applications. It provides security-by-design solutions for multi-cloud applications as well as solutions to their continuous security assurance. The latter is offered in form of a Software-as-a-Service solution named *MUSA Security Assurance Platform*.

The evaluation of MUSA DevOps and dynamic security assurance showed that the proposed methods and tools reduce the security flaws in the application implementation and ensure the multi-cloud application compliance to data protection requirements (including data integrity and confidentiality). The MUSA framework seeks the optimum way in which the multi-cloud application components will need to be created and interact in order to ensure a holistic management of security at runtime. It also advances over the state of the art in security-aware cloud SLAs, which foster clarity and transparency in cloud service provisioning. It also embeds security mechanisms into the application components obtaining as a result a self-protecting multi-cloud application at runtime, which facilitates continuous monitoring and dynamic security enforcement.

The MUSA Security Assurance platform enables cloud transparency by informing multi-cloud application providers on the real-time behaviour of both the application and the multiple cloud services underneath. Detected non-compliance with respect to security guarantees in the CSPs' and components' SLAs are early raised and corrected, or at least mitigated. As a result, the presented approach enables multi-cloud applications be smart, secure, and self-adaptive, increasing trust in cloud.

More evaluations of the MUSA framework are planned in the next months within the two case studies presented in section VI in order to assess its integration into a unique kanban based solution to enhance the collaborative work between the DevOps Team members.

Acknowledgment

The MUSA project leading to this paper has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 644429. We would also like to acknowledge all the members of the MUSA Consortium for their valuable help.

REFERENCES

[1] Rightscale. (2017) Cloud computing trends: 2017 state of the cloud survey. [Online]. Available: http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2017-state-cloud-survey

[2] Deloitte. (2017) Measuring the economic impact of cloud computing in europe, smart number: 2014/0031. [Online]. Available: https://ec.europa.eu/digital-single-market/en/news/measuring-economic-impact-cloud-computing-europe

[3] E. Rios, E. Iturbe, L. Orue-Echevarria, M. Rak, V. Casola *et al.*, "Towards self-protective multi-cloud applications: Musa–a holistic framework to support the security-intelligent lifecycle management of multi-cloud applications," 2015.

[4] M. E. project consortium. (2015) Multi-cloud secure applications. [Online]. Available: http://www.musa-project.eu

[5] G. I.-C. T. Forum, "Use cases and functional requirements for inter-cloud computing," in *Global Inter-Cloud Technology Forum, GICTF White Paper*, 2010.

[6] N. Grozev and R. Buyya, "Inter-cloud architectures and application brokering: taxonomy and survey," *Software: Practice and Experience*, vol. 44, no. 3, pp. 369–390, 2014.

[7] A. J. Ferrer, F. HernáNdez, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. M. Badia, K. Djemame *et al.*, "Optimis: A holistic approach to cloud service provisioning," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 66–77, 2012.

[8] M. Rak, S. Venticinque, T. Mhr, G. Echevarria, and G. Esnal, "Cloud Application Monitoring: The mOSAIC Approach," in *2011 IEEE Third International Conference on Cloud Computing Technology and Science*. IEEE, nov 2011, pp. 758–763. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-84857173913&partnerID=MN8TOARS http://ieeexplore.ieee.org/document/6133226/

[9] M. E. project consortium. (2013) Model-driven approach for design and execution of applications on multiple clouds. [Online]. Available: http://www.modaclouds.eu

[10] P. E. project consortium. (2013) A model-based cross-cloud development and deployment platform. [Online]. Available: http://www.paasage.eu

[11] E. Kamateri, N. Loutas, D. Zeginis, J. Ahtes, F. DAndria, S. Bocconi, P. Gouvas, G. Ledakis, F. Ravagli, O. Lobunets *et al.*, "Cloud4soa: A semantic-interoperability paas solution for multi-cloud platform management and portability," in *European Conference on Service-Oriented and Cloud Computing*. Springer, 2013, pp. 64–78.

[12] D. Zeginis, F. D'andria, S. Bocconi, J. Gorronogoitia Cruz, O. Collell Martin, P. Gouvas, G. Ledakis, and K. A. Tarabanis, "A user-centric multi-paas application management solution for hybrid multi-cloud scenarios," *Scalable Computing: Practice and Experience*, vol. 14, no. 1, pp. 17–32, 2013.

[13] D. Bernstein and D. Vij, "Intercloud security considerations," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 537–544.

[14] J.-M. Bohli, N. Gruschka, M. Jensen, L. L. Iacono, and N. Marnau, "Security and privacy-enhancing multicloud architectures," *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 4, pp. 212–224, 2013.

[15] K. Fatema, V. C. Emeakaroha, P. D. Healy, J. P. Morrison, and T. Lynn, "A survey of Cloud monitoring tools: Taxonomy, capabilities and objectives," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2918–2933, oct 2014. [Online]. Available: http://dx.doi.org/10.1016/j.jpdc.2014.06.007 http://linkinghub.elsevier.com/retrieve/pii/S0743731514001099

[16] A. Naser, M. F. Zolkipli, S. Anwar, and M. S. Al-Hawawreh, "Present Status and Challenges in Cloud Monitoring Framework: A Survey," in *2016 European Intelligence and Security Informatics Conference (EISIC)*. IEEE, aug 2016, pp. 201–201. [Online]. Available: http://ieeexplore.ieee.org/document/7870228/

[17] C. Zeginis, K. Kritikos, P. Garefalakis, K. Konsolaki, K. Magoutis, and D. Plexousakis, "Towards Cross-Layer Monitoring of Multi-Cloud Service-Based Applications," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, vol. 8135 LNCS, pp. 188–195. [Online]. Available: http://link.springer.com/10.1007/978-3-642-40651-5_16

[18] N. Ferry, A. Rossini, F. Chauvel, B. Morin, and A. Solberg, "Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems," in *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*. IEEE, 2013, pp. 887–894.

[19] SINTEF. (2013) Model-based provisioning and deployment of cloud-based systems. [Online]. Available: http://cloudml.org

[20] C. Zeginis, K. Kritikos, P. Garefalakis, K. Konsolaki, K. Magoutis, and D. Plexousakis, "Towards cross-layer monitoring of multi-cloud service-based applications," in *European Conference on Service-Oriented and Cloud Computing*. Springer, 2013, pp. 188–195.

[21] A. Brogi, A. Ibrahim, J. Soldani, J. Carrasco, J. Cubo, E. Pimentel, and F. D'Andria, "Seaclouds: a european project on seamless management of multi-cloud applications," *ACM SIGSOFT Software Engineering Notes*, vol. 39, no. 1, pp. 1–4, 2014.

[22] L. Columbus, "Computerworld's 2015 forecast predicts security, cloud computing and analytics will lead it spending," *Online at http://www.forbes.com/sites/louiscolumbus/2014/11/26/computerworlds-2015-forecast-predicts-security-cloud-computing-and-analytics-will-lead-it-spending/*, 2014.

[23] S. E. project consortium. (2015) Secure provisioning of cloud services based on sla management. [Online]. Available: http://www.specs-project.eu/

[24] I. Gartner, "Gartner it glossary devops," *Gartner IT Glossary*, 2017. [Online]. Available: http://www.gartner.com/it-glossary/devops

[25] S. O. Afolaranmi, L. E. G. Moctezuma, M. Rak, V. Casola, E. Rios, and J. L. M. Lastra, "Methodology to obtain the security controls in multi-cloud applications," in *Proceedings of the 6th International Conference on Cloud Computing and Services Science - Volume 1: CLOSER,*, INSTICC. ScitePress, 2016, pp. 327–332.

[26] V. Casola, A. D. Benedictis, M. Rak, and U. Villano, "A security metric catalogue for cloud applications," in *Complex, Intelligent, and Software Intensive Systems - Proceedings of the 11th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2017), Torino, Italy, July 10-12, 2017*, 2017, pp. 854–863. [Online]. Available: https://doi.org/10.1007/978-3-319-61566-0_81

[27] Openstack. (2015) Pacemaker cluster stack. [Online]. Available: https://docs.openstack.org/ha-guide/controller-ha-pacemaker.html

[28] OASIS. (2013) extensible access control markup language (xacml) version 3.0. [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html

[29] National Institute of Standards and Technology (NIST), "Security and privacy controls for federal information systems and organizations," vol. 800-53, pp. 1–460, apr 2013. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf