2017 IEEE 31st International Conference on Advanced Information Networking and Applications

# A process model to support continuous certification of cloud services

Immanuel Kunz and Philipp Stephanow
*Fraunhofer Institute for Applied and Integrated Security (AISEC), Munich, Germany*
*{firstname.lastname}@aisec.fraunhofer.de*

***Abstract*—Current research on cloud service certification is working on techniques to continuously, i.e. automatically and repeatedly, assess whether cloud services satisfy certification criteria. However, traditional certifications are conducted following static processes which are not designed to meet the requirements of continuous certification techniques. In this paper, we address this gap by redesigning the traditional certification process and adding suitable tooling to support continuous certification of cloud services. To that end, we analyze and generalize traditional certification processes and, on this basis, develop a novel, executable process model to detect ongoing changes of cloud services and adapt continuous certification techniques accordingly. We present our prototype which implements the process model and show how it allows us to automatically reconfigure continuous certification techniques according to changes observed in the target of certification as well as to continuously report certification results.**

***Keywords*-Cloud Certification; Certification Process**

## I. Introduction

*Continuous*, that is, repeated and automatic checks of certification criteria are subject to research as new certifications for cloud services are developed. However, integrating such continuous checks in existing certification processes is not trivial. These processes involve many tasks besides the actual checks and need to be properly modified to sufficiently support continuous certification techniques.

A certification process is the sequence of steps required by a certification schema, e.g., CSA STAR [1]. If the process is conducted successfully, then a *certificate* is issued, stating *compliance*. A certification process consists of many tasks: Choosing a certification schema, selecting auditors, paying certification fees etc. Only one of these steps contains the actual audit, i.e. mostly manual tasks conducted by professional auditors to assess whether a cloud service adheres to the certificate's requirements. Yet a cloud service's attributes may change over time and these changes are not predictable or detectable by a customer or even by the provider [2]. Thus auditing cloud services requires a different approach capable of continuously detecting ongoing changes and assessing their impact on certificates' requirements.

Automatically reasoning about certificates' requirements requires collecting and evaluating evidence [3], i.e. observable information of a cloud service. Recent research has proposed test-based [4][5][6] and monitoring-based [7][8][9] certification techniques to continuously evaluate a cloud service's compliance with certificates' requirements. *Monitoring-based* certification techniques use monitoring data as evidence collected from components involved in service delivery during the execution of the cloud service. *Test-based* certification techniques produce evidence by controlling some input to the cloud service and evaluating the output, e.g., calling a cloud service's RESTful API and comparing responses with expected results. However, these novel certification techniques impose new requirements on the certification process which are not supported by traditional certification processes. These requirements include the need to automatically reconfigure continuous certification techniques according to changes observed in the target of certification (TOC) as well as continuously report of certification results. Furthermore, conducting the certification process frequently and in short intervals bears further implications, requiring suitable tool support.

In this paper, we propose a novel certification process model to support continuous cloud service certification. To that end, we analyze process steps taken during eight popular certification schemas to derive a general model representing traditional certification processes. Based on this general process model, we develop a novel, executable certification process model which is capable of supporting continuous certification techniques.

Certification processes can be understood as business processes with specified inputs, a set of tasks and an output. To be able to analyze and manage business processes, we convert them into business process models, i.e. a formal representation that can be analyzed, executed and monitored. We create process models for certification processes using the business process modeling language Business Process Model and Notation (BPMN). Process models especially help dealing with complex processes as they provide insight and document procedures. As such we use them to redesign processes with the purpose of optimizing them [10].

The contributions of this paper are as follows:

- A reference process model of traditional certification processes,
- a novel, executable process model to support continuous certification of cloud services, and
- exemplary application of the novel process model.

Having described the traditional process model (Section II), we present our process model to support continuous certification of cloud services (Section III). Thereafter, we describe how to implement and apply our process model

1550-445X/17 $31.00 © 2017 IEEE
DOI 10.1109/AINA.2017.106

(Section IV) and evaluate the model (Section V). Finally, we discuss related work (section VI) and conclude this paper (Section VII).

## II. Traditional Certification Process

This section describes a model representing traditional certification processes specified by certification schemas such as CSA Star. After having presented roles involved in traditional certification processes (Section II-A), we describe a reference process which we obtained by inspecting eight current certification schemas (Section II-B).

### A. Participating parties

The participating parties in the process are the Cloud Service Provider (CSP), the certification agency (CA) and the auditors. The CSP is the client who wants to be certified by a CA to proof to (potential) customers that certain standards are met. The CA mandates a team of independent auditors to conduct the audit. For instance, the first task in a certification process is to choose a certification schema which is done by the CSP. Later, the auditors are chosen by the CA and they demand reference documents from the CSP and conduct test procedures.

### B. Reference process

To obtain a general representation of how certification is conducted as of today, we used eight different certification schemas to derive a reference process. Note that these certification schemas are exemplary and other schemas can be added as required in the future to extend the reference process model. It is a superset of all process steps taken during the certification of the following certification schemas: ISO 27001 [11], IT-Grundschutz [12], ISIS12 [13], ECSA [14], DCSA [15], SAP HANA [16], EuroPrise [17] and NIST 800-37 [18]. It is thus a general process that does not represent one specific certification schema but can be adapted to any of the ones it incorporates. On the one hand, the process steps are very similar for all certification schemas. For instance, filling out an application form and sending it to the CA, form part of almost every included schema. Other tasks, on the other hand, differ noticeably, e.g., a self-assessment is only supposed to be done in the ECSA and DCSA schemas.

The traditional certification process can be divided into the three phases *initialization*, *audit* and *certification*. Figure 1 shows the phases and states that the traditional certification process consists of. In general, it is a linear process in which the phases initialization, audit and certification are conducted sequentially and their respective states are entered as soon as they are finished. The state *Certified* in Figure 1 includes the variable $cycles$ which counts how many times the state *Certified* has been reached. When the process is in the state *Certified*[1], typically after one year a surveillance audit is conducted. Consequently, the process moves to the state *Audited* again and can be certified again. Furthermore, typically after three years time, a complete *Re-certification* is initialized and the process moves to the state *Initialized* again such that the whole process is executed again. This is why a surveillance audit is conducted when $cycles < 3$ and a re-certification is done when $cycles == 3$. In the following, the three phases are explained in more detail.

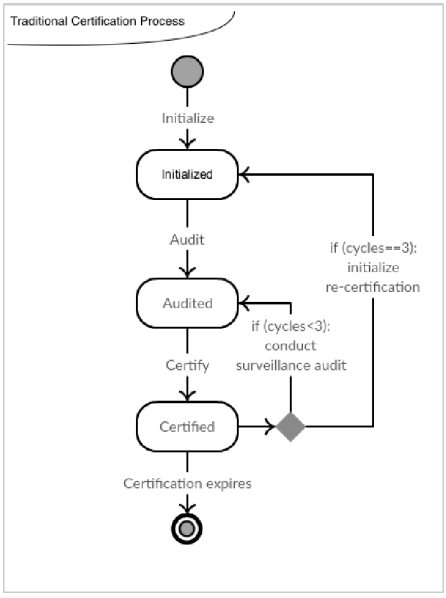


Figure 1: State diagram of the traditional certification process

*1) Initialization Phase:* The certification process begins with the initialization phase where all preparations are done in order to prepare the audit. It is initiated by the CSP who needs to choose a certification schema first to be certified for. Then, a self-assessment is done and a CA needs to be chosen to conduct the certification with. Later, the CSP needs to apply for the certification before a contract is made between CSP and the chosen CA. The CA then chooses independent auditors to conduct the audit. In the end of the initialization phase, the scope of the certification, i.e. the definition of the actual subject of the certification, is defined.

*2) Audit Phase:* The audit phase is the second phase and it includes all test procedures that need to be done by the auditors to proof that the certification criteria are fulfilled. The auditors begin with the examination of reference documents that the CSP provides and prepare the on-premise audit. These preparations include an inspection plan and the definition of control samples. Then, the on-premise audit is conducted and finally, results are reported to the CA. The CA has the responsibility to verify that the certification requirements have extensively been reviewed. The report is evaluated and reworks might be demanded.

*3) Certification Phase:* In the certification phase, the CSP pays the certification fees and the certificate is issued (or

[1]Note that for reasons of simplicity, we omit the possibility of unsuccessful certification in this paper's presentation.

refused) by the CA. Based on the report provided by the auditors, the CA makes a decision about the issuing of the certificate. If the decision is positive, the certificate can be published and is registered by the CA.

All in all, the traditional certification process is a static, linear process that includes many manual tasks. The traditional certification process as described in this section, was the basis for the redesign we propose in the next chapter. To create this redesign, we created a process model of it using BPMN 2.0. Creating a process model is advantageous for several reasons. Firstly, it is configurable which means that individual tasks can be modified or replaced. This was necessary as we wanted to make modifications to it. Secondly, it provides better overview of the process as a whole and is better manageable. Thirdly, a process model can be executed using a business process management platform (for further details see Section IV).

Analyzing the traditional certification process, we identified several steps that need to be automated in order to satisfy the requirements that arise when continuous certification techniques are applied. On this basis, we developed the continuous certification process.

## III. Continuous Certification Process

This section describes the certification process model to support continuous certification of cloud services. We begin with describing the requirements of continuous certification process model (Section III-A) and, on this basis, detail its design (Section III-B). Finally, we discuss the differences between the novel process model and the traditional certification process model (Section III-C).

### A. Requirements

When continuously assessing certification criteria, the traditional certification process cannot be used because automatic and reported checks require other automatic functions in the process that the traditional certification process does not provide. The impact of continuous certification techniques on the certification process is twofold:

1) *Changes of the TOC:* Cloud services are subject to ongoing changes which may affect operation and results of continuous certification techniques. Therefore, changes of the TOC need to be detected in order to adapt configurations of continuous certification techniques accordingly.
2) *Changes of certificates' status:* Continuous certification techniques are applied automatically and repeatedly, thus continuously producing results which may affect the status of a cloud service's certificates. Thus, results of continuous certification techniques as well as changes of certificates' status need to be reported continuously.

In the traditional certification process, however, neither automatic reconfiguration of the TOC nor continuous reporting of certification techniques' results are supported. For instance, test-based certification as proposed in [4] requires such functions as it assumes a predefined TOC. Furthermore, its functionality is required to handle produced test results.

We propose a continuous certification process to bridge this gap. Regarding the first requirement, a mechanism to automatically reconfigure certification techniques according to changes of the TOC, we introduce a new phase to the certification process which we call *scoping*. It is executed after the initialization phase and provides the audit phase with the current scope of the TOC. Furthermore, to meet the second requirement, a mechanism to continuously report last results of certification techniques, we introduce a new subprocess to the certification phase. In this subprocess, results provided by, e.g., test-based certification techniques are processed and the certificate is updated.

Besides these requirements, there are implications to be considered for the certification process itself if it is conducted continuously. While the traditional process is a static process that is meant to be conducted once, the continuous certification process is executed as follows: It starts with the *Initialization* phase. This involves all preparations required to start the continuous certification techniques, including automatic definition of the TOC, test- and monitoring-based techniques and continuous reporting of results. Afterwards, the *Scoping*, *Audit* and *Certification* phase are executed continuously, i.e. automatically and repeatedly.

Figure 2 shows the states and phases of the continuous certification process model. After its start, the initialization phase is conducted and the process moves to the state *Initialized*. In this state the automated phases scoping, audit and certification are conducted, moving through their respective states sequentially. Finally, when the timer runs out, the scoping is started again. Otherwise, it can be terminated. The phases *Initialization*, *Audit* and *Certification update* are explained in more detail hereafter.

### B. Design

In the continuous certification process, we add one phase to the traditional process model. As stated earlier, it is necessary to execute the scoping, i.e. the definition of the TOC, continuously. Therefore, the scoping is separated from the initialization phase. The initialization phase is only conducted once to prepare the following process steps.

*1) Initialization Phase:* The continuous certification process begins with the initialization phase which differs from the one in the traditional process only in one point: The scoping is not part of it anymore but is replaced with a new step which is to set up all systems that are necessary for the continuous certification process, i.e. the three phases *Scoping*, *Audit* and *Certification* that are executed continuously. This set up can include, e.g., installing an inventory

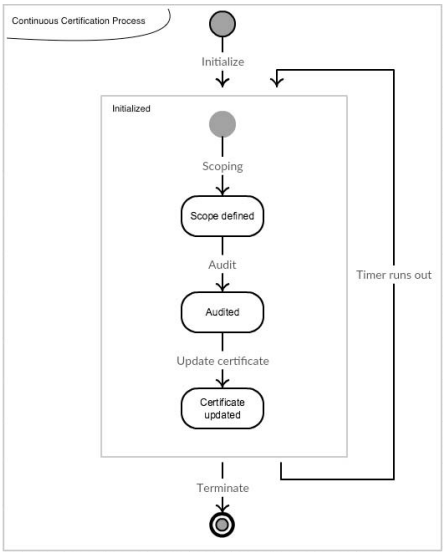

Figure 2: State diagram of the continuous certification process

management system and modifying the process models to fit the desired certification schemas.

*2) Scoping Phase:* In this phase, the TOC is defined. The scoping mechanism we propose works on one important assumption which is that the CSP is running an inventory management system (IMS). Such a system continuously gathers information about the assets that are present in a network. Thus, we assume that the CSP runs an IMS that provides information about the infrastructure layout of the provider required to define the TOC. Such information may consists of an overview of the current network including hardware specifications, network addresses, connected devices etc. Furthermore, we assume that it is part of the certification agreements between the CSP and the CA that the CSP makes inventory reports generated by an IMS available to the continuous certification process.

Figure 3 shows the BPMN process model we created for this phase. It is initiated by a timer that triggers the process every time a specified time interval has passed, that is, the process execution is scheduled. However, it can also be implemented event-driven. For this purpose, we included a second start event in the model which is a *Message Event* that can be used to start a process execution any time, e.g., via a call to the REST-API.

As soon as a process execution is triggered, the first task is to check whether a new inventory report has been provided by the CSP. In the exceptional case that no new inventory is available, the *Exclusive Gateway* directs the process execution to a *Mail Task* that sends a notification to the CA so further steps can be taken manually. Hereafter, an *End Event* terminates the process instance. However, if a new inventory report is available, the process continues with comparing the file with the previous inventory to determine whether it has changed. Depending on the agreement that CSP and CA made, such changes can consist of newly added network devices or changed hardware specifications. If the inventory has changed, a new scope is computed and updated. In this step, the inventory report is further manipulated. This is necessary as there may be, e.g., devices in the inventory not subject to the certification and therefore need to be excluded from the scope. Furthermore, the definition of the scope can be used to modify the inventory report such that it only contains the information necessary for the monitoring-based and test-based certification techniques. This means unnecessary or potentially sensitive information can be deleted while other information is included. Finally, control samples, e.g., network addresses, are chosen randomly from the current scope and are passed on to the audit phase.

*3) Audit Phase:* The third phase is the audit phase where any test-based and monitoring-based certification techniques are deployed, each of which produce results indicating whether a cloud service complies with requirement of the certificate. These techniques are configured according to the selected control samples. The discussion of these certification techniques is not in focus of this paper.

*4) Certification Phase:* In the certification phase, certification fees are payed and the audit results are reported. We assume that in a continuous certification, there is not a one-time fee anymore but they are payed, e.g., monthly. For the reporting, different approaches are possible. We propose a push-based approach that automatically updates the certificate on a public platform. The reliable and secure implementation of this reporting mechanism is important as it can have influence on the CSP's reputation. Consider, for instance, technical problems occur during the audit phase which result in false negative results, i.e. a certification technique incorrectly indicates that the TOC is not satisfying certificate's requirements. Publishing such incorrect results instantaneously may severely damage the CSP's reputation.

Continuously reporting certification results thus requires not only to report a *valid* or *invalid* certificate but also – as Anisetti et al. [5] propose – to allow for degrees of validity and to include an exceptional case if results of the continuous certification techniques appear to be erroneous.

### C. Comparison

In this section, we compare the traditional and continuous certification process models based on the properties *content*, *size* and *complexity*.

*1) Initialization Phase:* In the initialization phase, only the last task of the traditional process has changed to model the continuous certification process. It does not contain the definition of the scope anymore since this task is excluded from the initialization phase. In turn, the scope becomes a novel, second phase of the continuous certification process while the last step of the initialization phase of the continuous process model consists of setting up and initialize

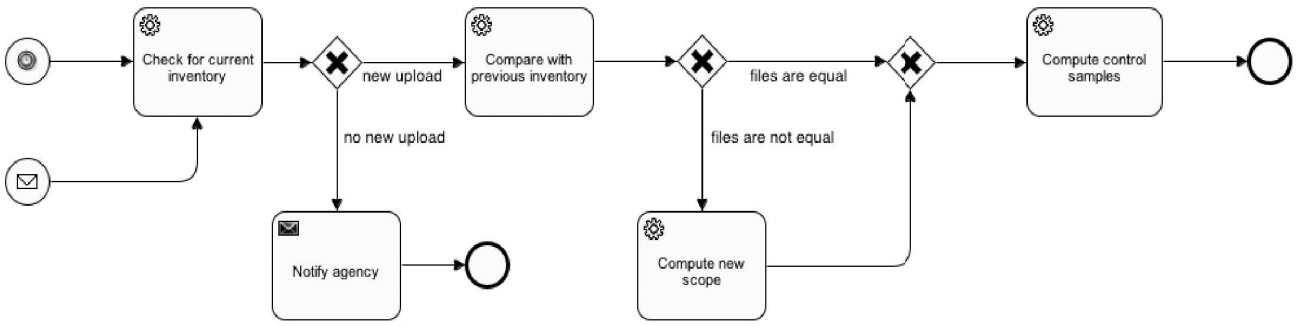


Figure 3: BPMN model of the scoping phase of the continuous certification process

further steps required during the continuous certification.

*2) Audit Phase:* The audit phase of the continuous process model now contains the continuous certification techniques instead of a one-time on-premise audit. Also, in the continuous process model, defining control samples is not part of the audit phase anymore. It is instead moved to the scoping because this step rather belongs to the definition of the scope than to the actual audit methods.

*3) Certification Phase:* The certification phase is significantly different as well. In this model, decisions about the certificate's state are made automatically. The continuous reporting updates the certificate immediately after the continuous certification techniques provided new results. Furthermore, the payment of certification fees needs to be changed in a continuous environment.

In general, the continuous process is more coherent than the traditional one as less coordination between the participants is needed. This conclusion can be derived when comparing the BPMN process models of the traditional and the continuous certification processes[2]: Both use so called *Message Events* whenever communication between the participants is needed, for instance, when the auditors send their report to the CA. *Message Events* trigger their following process step as soon as a pre-defined message is received via the REST-API. In contrast to the continuous process model, the traditional process model comprises many small processes that interact via *Message Events*. In total, the traditional model contains 20 *Message Events* whereas the continuous one only contains 10 *Message Events*. Consequently, in the continuous model, less communication is needed. While *Message Events* do not necessarily represent a need for human communication, a smaller amount nevertheless indicates less interacttion and therefore less complexity.

The continuous process model is, however, more modular. While the traditional process is one coherent process model, the continuous one consists of four independent submodels that fulfill different goals, e.g., scoping and reporting. The continuous process model uses *Message Events* primarily to control the execution of these individual submodels.

Lastly, a comparison between the two process models on the basis of other metrics, e.g., controllability, is not appropriate because they are designed for different purposes. The continuous certification process model is supposed to guide a largely automated certification process whereas the traditional model's primary purpose is to provide a formal representation of the reference process which serves as the basis to develop the continuous process model.

## IV. Application

In this section we describe how the proposed certification process can be applied. After having specified our setup and environment (Section IV-A), we show how the process model can be executed (Section IV-B). Then we describe an exemplary scenario (Section IV-C) and finally discuss how our approach can be applied to allow for incremental transition from traditional certification to continuous certification (Section IV-D).

### A. Setup and Environment

To implement and execute the proposed process model for continuous certification, we used the business process management platform *Activiti*[3]. It is an open source BPMN 2.0 process engine framework that provides an environment for running business and technical processes. It consists of several components which include the BPMN 2.0 Modeler that is available as a plug-in for the Eclipse IDE. We used this component to create the process models. Also, it includes the *Activiti Engine* which is the core workflow processor and the *Activiti Explorer* which is a web application that can be used to monitor the process executions. For instance, it can be used to observe which task is being executed and to manage and analyze data about current and finished process instances. Furthermore, it provides native support for the *Spring Boot*[4] framework which we used as well. As described earlier, the scoping phase of our

[2]Due to space limitations, presentation of the complete BPMN process models is omitted in this paper.

[3]http://activiti.org/

[4]https://projects.spring.io/spring-boot/

process model requires an inventory report provided by an IMS. To that end, we used the *OCS Inventory NG*[5] which, among others, creates an overview of devices in a company's network and exports this report as a CSV file.

### B. Process execution

Executing a process model with Activiti means that every activity in the model is executed in the designated sequence as defined by BPMN 2.0. In Activiti, tasks can be assigned to humans but they also can be assigned to a program code. Considering the submodels we propose for the scoping, audit and certification phase of the continuous certification process model, almost all tasks are so called *Script Tasks* that call Java methods. This way, these three phases work fully automatically.

The execution of the scoping phase is scheduled by a *Timer Event* that triggers the execution in our implementation on a daily basis. However, its execution can also be triggered manually. The *End Event* of the scoping phase is configured to trigger the execution of the audit phase. The *End Event* of the audit phase in turn triggers the execution of the certification phase. This way, each phase is executed as soon as the previous one is completed. That way, each phase uses the latest information. For instance, the audit phase is executed with the latest scope and the certification phase is executed with the latest test-based and monitoring-based certification results.

Moreover, it is also possible to schedule the execution of all three phases individually. This can be desirable because the scoping phase might not finish as intended if, e.g., the inventory report cannot be processed. In this case, if triggered strictly sequential, then the audit and certification phase would not be executed at all. However, if the executions of all three phases are scheduled individually, it is necessary to verify in the beginning of each phase that data to be used, e.g., the scope or the test report, is up to date.

### C. Exemplary scenario

To illustrate how the proposed system works in a real scenario, consider the following use case: The underlying standard of a certain certificate demands that a service is available at all times, e.g., IVS-04 of the Cloud Control Matrix (CCM) [19] upon which the CSA certificate is based [1]. The auditors want to apply test-based certification techniques to evaluate the availability of the cloud service. Following the continuous certification process, in the initialization phase respective systems are installed and configured. Then, in the scoping phase, the information from the IMS is used to determine which components of the inventory are used to deliver the cloud service and thus should be considered when assessing the availability of the service. The scope is computed accordingly and control samples are selected. These samples are passed on to the audit phase where test-based techniques' configurations are adapted accordingly. The certification techniques produce results which are in turn used to generate a report which is passed on to the certification phase. Here, the certification techniques' results are evaluated and the evaluation is published on a public platform.

[5]http://www.ocsinventory-ng.org

### D. From traditional to continuous certification

The continuous certification process model and its submodels are made with the purpose of supporting a continuous certification of cloud services. However, it is also possible to use the traditional process model to guide a traditional certification, making the traditional process partly executable. This way, the model of the traditional process can support auditors and other participants in the traditional process. This makes the traditional process execution transparent and controllable. The process model can be deployed and made accessible to the participants as soon as the certification is initiated. If there is a detailed model for the specific certification schema that is being followed, the auditors can use it to, e.g., show that certain milestones have been achieved and can share important documents to proof to the CSP and the CA that the audit is conducted correctly.

This approach paves the way to transition from the traditional to the continuous process step by step. Besides the traditional process and the continuous one, there are several intermediate levels of process automation that can be achieved incrementally. One step would be to conduct the certification using a model-based approach where the traditional certification is conducted following the BPMN model. In this case, auditors may provide important documentation to all certification process' stakeholders through the executable process model.

## V. Evaluation

In this section, we evaluate our process model for continuous certification present in Section III. To that end, we draw on the requirements *Configurability*, *Abstraction*, *Robustness* as well as *Automation and Controllability* which were introduced in Section III-A.

### A. Configurability

In general, BPMN models are configurable as they can be modified, e.g., via the Activiti Modeler or Designer plug-in for Eclipse simply by dragging, adding and deleting the shapes and connecting them with the sequence flow arrows. This is done without having to configure the model any further. Also, the models in Activiti are defined as XML files. As such, a configuration can also be edited by other applications, e.g., using text editor macros.

The configurability is not only important for the adaptability of the process model to certain certification schemas. It is also important to facilitate the implementation of new

activities into the process or to modify existing ones. This was necessary when implementing the proposed changes of the traditional certification model to support continuous certification.

### B. Abstraction

The models for the different phases present different degrees of abstraction. Since the initialization phase is not changed, except that the scoping is replaced, the degree of abstraction for this phase is not changed either. As such, it does not differ from the traditional process model which does provide a suitable degree of abstraction as it consists of all tasks from the selected certification schemas (see Section II-B) and as such is adaptable to any of them but still presents very specific steps to answer relevant questions about the process.

The automated scoping is a rather specific model and hence has a low degree of abstraction. This is the case because it is created as a specific implementation that uses inventory reports of IMS such as OCS Inventory NG.

The reporting of continuous certification techniques' results presents a rather high degree of abstraction as it does not specify how monitoring-based or test-based certification results are updated. That way, our certification subprocess can support results of any continuous certification technique.

### C. Robustness

Robustness is an important quality of a business process model because it means that exceptions and unexpected behavior can be dealt with. Consequently, the need to monitor its execution is reduced.

For the proposed model two types of robustness can be distinguished. Firstly, robustness against unexpected behavior in the process logic. This means that conditions that are assumed to be fulfilled, are not. This can be the case if, for example, there is no new inventory report available in the scoping subprocess. In this case the process cannot continue as no new scope can be computed for the successive audit phase. At several points, the model deals with unexpected events like these by forking the sequence flow to a task that sends a notification to an auditor or the CA before ending the process instance. This way, a responsible person is informed about the problem without having to monitor the process execution. Secondly, there are technical errors that can occur in the Java code that is called by the process tasks or in the Activiti Java code. If, for example, a process variable is written to the database but it does not implement the Serializable interface, a rollback will happen in the database which can make important process results invalid. Thus, these errors depend on how the code is implemented.

Other scenarios are possible in which errors can hinder the process execution. For instance, if a new inventory report is provided but it is not readable. In this case, the Java method would throw an error and the process instance would end. Another scenario is that the inventory report is readable but does not contain any entries or very few. If this is not considered in the code, it could happen that an empty inventory report is accepted and the audit returns a positive result although no component or only very few were subject to the continuous certification techniques. Similar scenarios have to be taken into account when considering computation of scope and the continuous reporting of certification results.

### D. Automation and Controllability

Furthermore, it is defined that the models should provide a high degree of automation but should still keep the possibility to control the process manually. These two requirements are in conflict with each other as human interaction always hinders the automatic execution of the process. In the proposed model, controllability is achieved by modularity, i.e. the separation of the process steps scoping, monitoring and reporting. These individual models can all be initiated manually using commands to the REST-API. In addition, also the certificate's public status can be changed manually via such commands. Still, the model presents a business process that works fully automated under certain assumptions, e.g., that an IMS is used, proposing a suitable solution for the conflicting requirements automation and controllability.

## VI. Related Work

Cimato et al. first introduce the notion of cloud service certification [3], including initial remarks on how to design the process of cloud service certification. Anisetti et al. [4][5] introduce a test-based certification scheme for cloud service. Krotsiani et al. explore monitoring-based certification techniques [8][20]. Katopodis et al. [21] advocate for *hybrid* certification techniques combining testing-based and monitoring-based evidence since monitoring or testing alone can only cover parts of a cloud service's behavior. Yet, none of the above approaches address the challenge of adapting the traditional certification process to support continuous certification techniques, let alone providing a solution.

When executing a continuous certification process, monitoring-based and test-based certification results have to be published continuously. For this purpose, [22] include a *Certificate Generator* in their certification framework which stores a certificate in a special database with restricted access. This approach, however, is different from the one proposed here since our reporting mechanism does not include any access restrictions to the final test result.

[23] propose a certification process which consists of the four phases *Monitor-Analyze-Plan-Execute* which are, similar to our continuous certification process, executed sequentially and automatically. However, this process also assumes a given TOC. Furthermore, instead of a reporting of certification results, it includes a functionality to maintain the certificate's validity by adapting to misconfigurations of the TOC.

In general, existing works follow a rather integrative approach which does not separate the scoping and reporting from the assessment of certification criteria. A separation, however, is done in this work because it enables controlling and monitoring of the process.

## VII. Conclusion

In this paper, we presented a process model to support continuous certification of cloud services. We started out with modeling a reference process based on traditional certification schemes and redesigned several steps of that traditional process to support automatic reconfiguration of test-based and monitoring-based certification techniques as well as to continuously report certification results.

As part of future work, we will investigate how to handle process violations, e.g. unavailable or incorrect inventory reports. Furthermore, there are several security issues that require attention, e.g., how to brace against attacks aiming at disrupting process execution. The latter is particularly challenging when our proposed process model is deployed in a distributed setting where process components are run by different participating parties like the CSP and the CA.

## Acknowledgment

This work was partly funded by the Federal Ministry of Education and Research of Germany, within the project *NGCert* (http://www.ngcert.eu), Grant No. 16KIS0075K, and by the EU H2020 project *EU-SEC*, Grant No. 731845.

## References

[1] Cloud Security Alliance (CSA), "Security, Trust and Assurance Registry (STAR)." https://cloudsecurityalliance.org/star/certification/.

[2] B. S. Kaliski Jr and W. Pauley, "Toward risk assessment as a service in cloud environments," in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, pp. 13–13, USENIX Association, 2010.

[3] S. Cimato, E. Damiani, F. Zavatarelli, and R. Menicocci, "Towards the certification of cloud services," in *9th World Congress on Services (SERVICES)*, pp. 92–97, IEEE, 2013.

[4] M. Anisetti, C. A. Ardagna, E. Damiani, F. Gaudenzi, and R. Veca, "Toward Security and Performance Certification of OpenStack," in *8th International Conference on Cloud Computing (CLOUD)*, IEEE, 2015.

[5] M. Anisetti, C. Ardagna, F. Gaudenzi, and E. Damiani, "A certification framework for cloud-based services," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC)*, pp. 440–447, ACM, 2016.

[6] P. Stephanow, G. Srivastava, and J. Schütte, "Test-based cloud service certification of opportunistic providers," in *9th International Conference on Cloud Computing (CLOUD)*, IEEE, 2016.

[7] M. Krotsiani, G. Spanoudakis, and K. Mahbub, "Incremental certification of cloud services," in *7th International Conference on Emerging Security Information, Systems and Technologies (SECURWARE)*, pp. 72–80, IARIA, 2013.

[8] M. Krotsiani and G. Spanoudakis, "Continuous Certification of Non-repudiation in Cloud Storage Services," in *13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 921–928, IEEE, 2014.

[9] P. Stephanow and N. Fallenbeck, "Towards continuous certification of Infrastructure-as-a-Service using low-level metrics.," in *12th International Conference on Advanced and Trusted Computing (ATC)*, IEEE, 2015.

[10] W. Van Der Aalst, *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media, 2011.

[11] International Organization for Standardization (ISO), "ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems — Requirements." https://www.iso.org/obp/ui/#iso:std:iso-iec:27001:ed-2:v1:en. Retrieved 10-2016.

[12] German Federal Office for Information Security (BSI), "IT-Grundschutz-catalogues, 13th version." https://www.bsi.bund.de/EN/Topics/ITGrundschutz/ITGrundschutzCatalogues/itgrundschutzcatalogues_node.html. Retrieved 10-2016.

[13] Bavarian IT Security & Safety Cluster, "Information security management: Isis12 – information security for small and medium-sized businesses,"

[14] EuroCloud Deutschland, "ECSA EuroCloud Star Audit Data Privacy Audit Guide." https://staraudit.org/fileadmin/userdaten/emaillinks/ecsa-data-privacy-audit-guide.pdf. Retrieved 10-2016.

[15] eco — Association of the Internet Industry, "About DCSA." https://www.dcaudit.com/about-dcsa.html. Retrieved 10-2016.

[16] SAP, "SAP Outsourcing Operations Provider." http://global.sap.com/community/ebook/2012_Partner_Guide/files/SAP_OutsourcingOperations_Provider_Certifications.pdf. Retrieved 10-2016.

[17] EuroPriSe GmbH, "EuroPriSe Criteria Catalogue," *https://www.european-privacy-seal.eu/EPS-en/Product-and-Service-Privacy-Certification*. Retrieved 10-2016.

[18] National Institute of Standards and Technology (NIST), "Guide for Applying the Risk Management Framework to Federal Information Systems," *Special Publication 800-37*, 2010.

[19] Cloud Security Alliance (CSA), "Cloud Control Matrix: Security Controls Framework for Cloud Providers & Consumers." https://cloudsecurityalliance.org/research/ccm/, 2013.

[20] M. Krotsiani, G. Spanoudakis, and C. Kloukinas, "Monitoring-Based Certification of Cloud Service Security," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pp. 644–659, Springer, 2015.

[21] S. Katopodis, G. Spanoudakis, and K. Mahbub, "Towards hybrid cloud service certification models," in *International Conference on Services Computing (SCC)*, pp. 394–399, IEEE, 2014.

[22] M. Egea, K. Mahbub, G. Spanoudakis, and M. R. Vieira, "A certification framework for cloud security properties: the monitoring path," in *Accountability and Security in the Cloud*, pp. 63–77, Springer, 2015.

[23] C. A. Ardagna, R. Asal, E. Damiani, N. El Ioini, C. Pahl, and T. Dimitrakos, "A Certification Technique for Cloud Security Adaptation," in *International Conference on Services Computing (SCC)*, pp. 324–331, IEEE, 2016.