Bryce *Journal of Cloud Computing: Advances, Systems and Applications* (2019) 8:23
https://doi.org/10.1186/s13677-019-0148-5

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

Security governance as a service on the cloud

Ciarán Bryce

Abstract

Small companies need help to detect and to respond to increasing security related threats. This paper presents a cloud service that automates processes that make checks for such threats, implement mitigating procedures, and generally instructs client companies on the steps to take. For instance, a process that automates the search for leaked credentials on the Dark Web will, in the event of a leak, trigger processes that instruct the client on how to change passwords and perhaps a micro-learning process on credential management. The security governance service runs on the cloud as it needs to be managed by a security expert and because it should run on an infrastructure separated from clients. It also runs as a cloud service for economy of scale: the processes it runs can service many clients simultaneously, since many threats are common to all. We also examine how the service may be used to prove to independent auditors (e.g., cyber-insurance agents) that a company is taking the necessary steps to implement its security obligations.

Keywords: Security, Security as a service, Compliance, Cloud, Process modeling, Burden of proofs

Introduction

Cyber-security is a major concern for today's economy and society. It is a particular problem for small companies since they can lack the means to defend themselves [16]. First, they might not be technically savvy enough to understand the security problem and understand the measures that have to be taken. Second, small companies might not be able to invest in security solutions from their working capital. Third, their IT infrastructure is often very minimal, for instance, many small companies still manage their client database, containing client personal data and credit card information, with desktop spreadsheet software. This all makes small companies particularly vulnerable to cyber-security attacks.

Compliance, which is the process of respecting government regulation in the operation of business, is another factor of risk for companies. Obviously, compliance regulations are enacted in the interests of consumers and society, but the implementation of compliance within a company is costly, as is the failure to be compliant. The European Union's General Data Protection Regulation (GDPR) on the protection of personal data is a good example of the challenge. The GDPR applies to all companies, irrespective of their size and business sector, and requires companies to take specific protective measures for client personal data. Repeated failure to adhere to GDPR can incur severe penalties, e.g., up to 4% of annual turnover. Like for security, small companies are struggling to conform to GDPR [6]. It is useful to tackle security and compliance together as both are concerned with controls on information handling.

Security Governance deals with the definition and implementation of processes to mitigate security risks. In an era of zero-day exploits, companies must accept that attacks will happen and some will succeed, so governance processes deal equally with prevention (security software installation, training, etc.), detection (data leaks in the Dark Web, etc.) and recovery (password resets, informing clients of data leaks for GDPR, etc.). Security governance also applies to protection of physical assets from theft or damage [14].

This paper presents the design and implementation of a security governance service. Processes are automated on a cloud service on behalf of client companies. The cloud is chosen since the processes are put in place by a security expert, and processes run in a secure infrastructure independent of the client. Further, for economy of scale, processes may run on behalf of many clients. The model

Correspondence: ciaran.bryce@hesge.ch
University of Applied Sciences and Arts of Western Switzerland, Geneva School of Business Administration – HES-SO, 1227 Geneva, Switzerland

presents a company with a dashboard showing risk for their company. Risk-mitigating processes are manually or automatically started in response to detected threats.

A particular feature of our approach is that processes may request *attestation* evidence from clients to show that they are following the steps requested by the processes. Evidence is collected at runtime, and can be used by external parties like auditors, insurance providers, etc. Thus, even in the event of a successful security attack at the client site, the client can still show that steps to prevent the attack had been taken.

The remainder of this paper is organized as follows. "Approach to security governance" section motivates the design of the governance model. Some processes are presented in "Governance processes" section, and the model's implementation in "Implementation" section. Related work is presented in "Related work" section. Conclusions and future work are discussed in "Conclusions" section.

Approach to security governance

The objective of the security governance model is, following the identification of assets, to implement processes to secure these assets. This approach to security is expounded by frameworks like the IEC/ISO 27000 series [5]. In our case:

- The model implements a risk management service that permits a company to track security risks and instructs on actions to take to mitigate risk. The service is managed by a trusted third party. This party can follow risk in real time, and like an asset manager surveying a financial portfolio, trigger processes to alter overall risk.
- We seek to automate governance processes as much as possible – several examples are presented in "Governance processes" section. Process automation in the enterprise environment is known to reduce errors and increase efficiency [7].
- Permit **attestation** of the implementation of risk-mitigating processes to third parties such as government auditors verifying compliance or to insurance agents verifying that contractual actions are being taken. An example of the latter might be the installation of anti-virus software that is required by a cyber-protection insurance policy.
- Exploit commonalities in risks faced. For instance, all companies face risks relating to malware or reputation, and the remedial actions are the same. This enables the trusted third party to reuse the same processes for many clients.
- Be simple for small companies to use, and require no special technical expertise on their part. Moreover, the processes must represent a range of risks faced by such companies. Many different risks have an equally fatal impact on the company. For instance, for a hair salon we contacted considered risk linked to data loss to be as important as risk linked to malfunction of hair-drying equipment. Small firms appreciate a package solution for risk, rather than having to delegate different risks to different consultants.

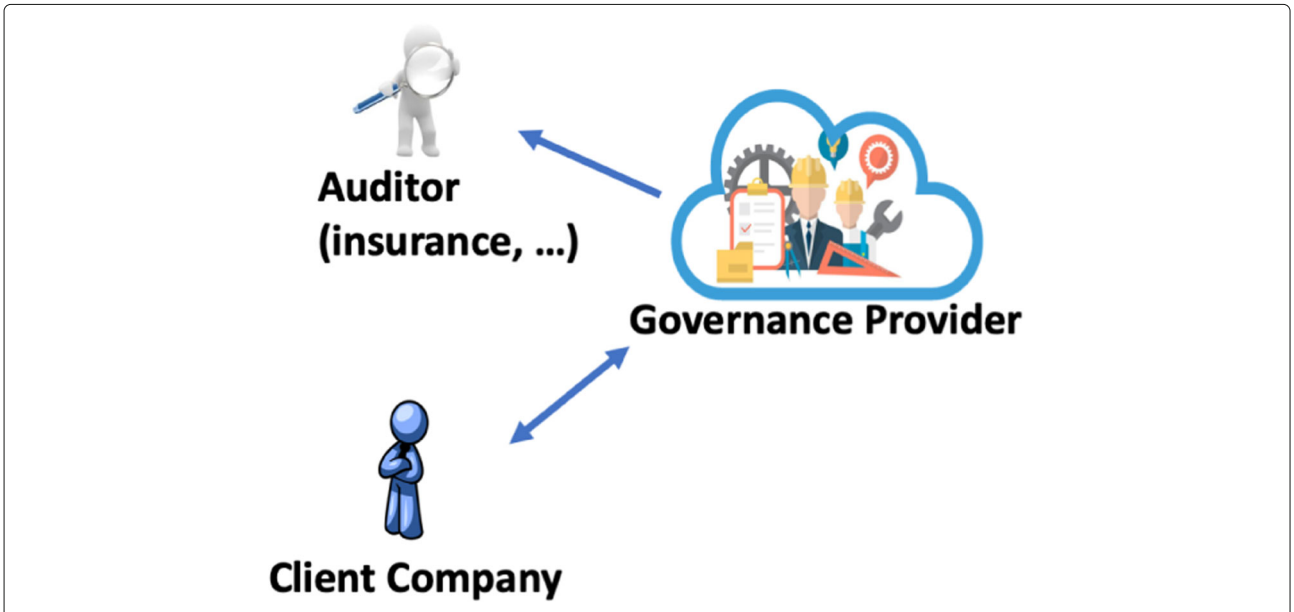

Fig. 1 Actors of the governance model. The arrows denote data flows. The governance provider offers a trusted environment in which it tracks risk for the client company and furnishes proof of implementation of risk-mitigating actions to auditors

Asset	Threat	Vulnerability	Risk	Trigger condition	Process Action	Risk
Cabinet and cash register	Theft	Work on ground floor	Max	2 hours since check by guard or theft in neighborhood	Security guard does round	L
			H	End of day	Lock cabinet	L
PC with client data	Data loss	Virus	H	Manuel or notification by provider of new virus	Install or update anti-virus	L
		Failure	M	1 day since last backup	Backup	Min
		GDPR failure	H	Data breach detected	Inform management & clients	Min
Reputation	Bad on-line review	Local review web-sites	H	Notification of review or time since last check	Scour local sites for reviews and reply if needed	Min

Fig. 2 Simple dashboard example

Actors

As illustrated in Fig. 1, there are three roles in the model: *clients*, *governance providers* and *auditors*.

A *governance provider* hosts the risk management service for client companies. He is an expert in risk and helps the client to identify assets, threats, vulnerabilities and actions to undertake to mitigate risks. This endeavor is facilitated by the fact that many risks are shared by all clients, e.g., theft, ransomware virus, etc., even if the risk level differs between clients. The provider runs a service that tracks the risk level of the client.

The *client* is any small or micro company that avails of the provider's risk management service. He is not assumed to have any expertise in security or compliance. Apart

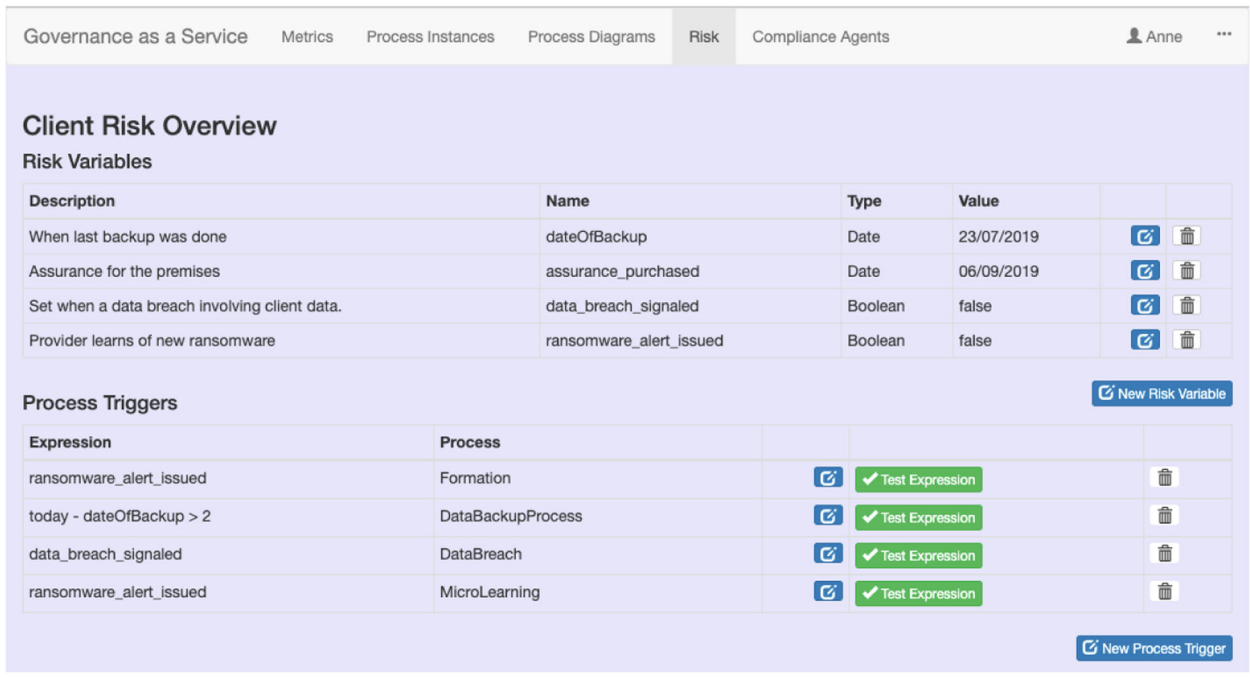

Fig. 3 Panel of the Governance Service

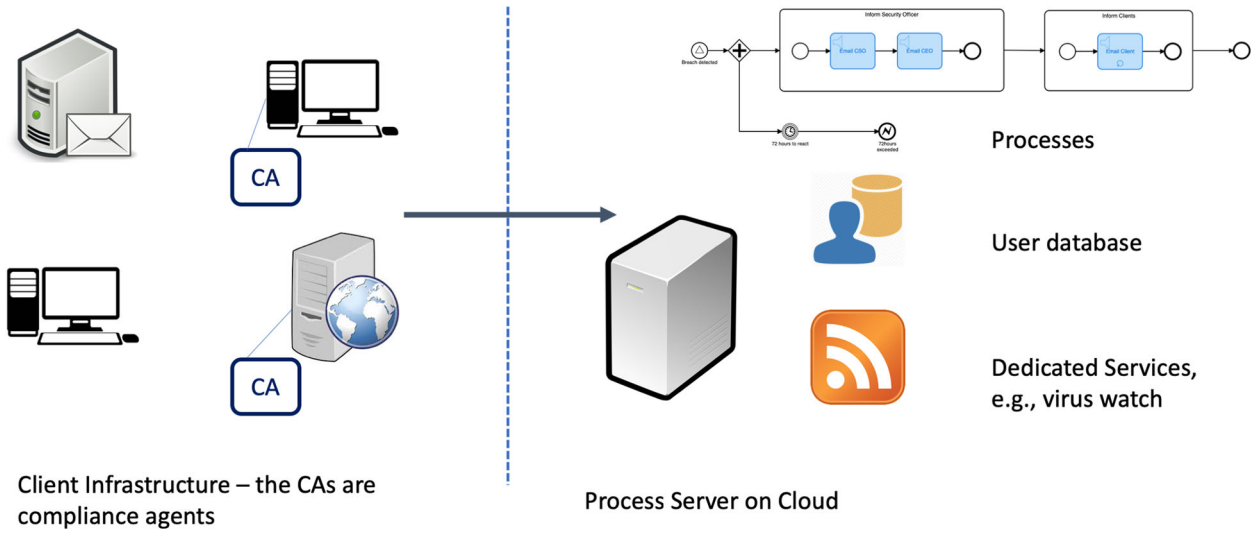

Fig. 4 Processes run on engine in a trusted environment

from regular discussions with the provider about the assets, threats and risks, his interaction with the service is limited to receiving alerts and instructions, and filling out Web-forms.

For each client, there may be distinct actors who fulfill different roles within the company, and who therefore execute different tasks. For instance, the office manager has different responsibilities to a security guard. Both execute different tasks of the client: the office manager executes tasks relating to defense (e.g., install anti-virus, etc.), the security guard is responsible for doing rounds, verifying the clean desk policy, etc.

An *auditor* is any independent party who wishes to see the status of the implementation of risk-mitigating actions by the client. In practice, this role is taken by a government auditor wishing to verify the implementation of compliance actions within the company, or by an agent of a cyber-insurance company who wishes to verify that prescribed risk-mitigating processes were implemented by the client following a claim made on its behalf.

Service components

There are two parts to the service offered by the governance provider. A **dashboard** front-end presents the client with an overview of the current risk state and actions to execute; a back-end **engine** automates the risk-mitigating actions for the client.

Dashboard

An example of an initial dashboard defined for a small company is illustrated in Fig. 2. Please note that for reasons of brevity, several assets and risks are omitted. The dashboard has an easy-to-interpret tabular format. Following an ISO 27000 like approach [5], and with the help of the provider, the client defines the assets he wishes to protect in the first column. Identified threats are listed in the second column and vulnerabilities that make these threats credible are listed in the third column.

A *risk level* value along with some *risk condition* that explains the risk level are given in columns 4 and 5. A risk mitigating process, composed of actions, for the threat-

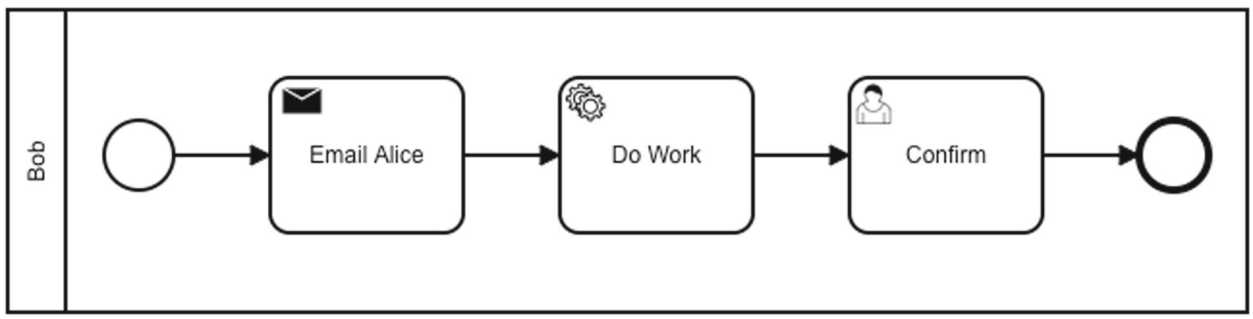

Fig. 5 Simple BPMN process

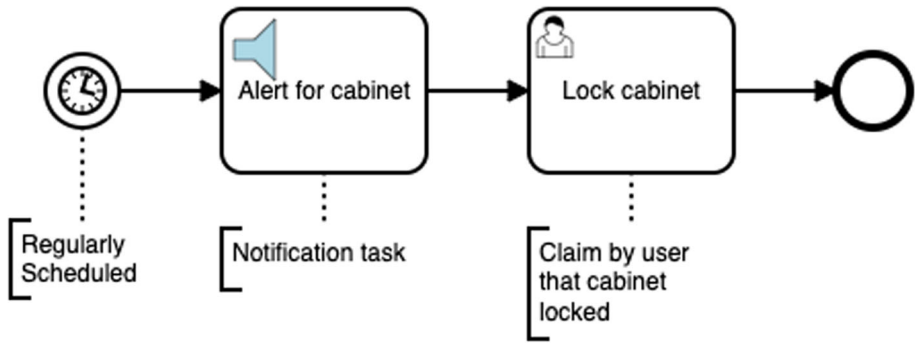

Fig. 6 Process for cabinet locking

vulnerability is mentioned in column 6 and the impact of execution of the process on the risk level is given in column 7.

The table is deliberately descriptive in a first step because it represents the level of detail employed in an interaction between a client and governance provider. The trigger conditions and the actions are programmatically defined in a second stage by the provider. The result is shown in Fig. 3 – a screenshot from our service.

The dashboard in Fig. 3 contains the risk variables defined for the client. Two classes of variables can be defined:

- **Client-related**, e.g., assurance_purchased, data_breach_signaled, etc. These variables can only be updated by processes deployed on the engine.
- **Provider** variables relating to some service that the provider handles, e.g., ransomware_alert_issued for a service that tracks CVEs for malware. The maintenance of these variables is the responsibility of the provider. He may change the value of these variables which can trigger a process as a consequence.

Engine

The engine runs in an isolated and trusted environment, under the supervision of the provider. The engine executes programmed actions for notification, confirmation by clients or custom tasks. Access control ensures that a process action for a client may only be executed by that client. The environment stores client risk values and these may not be directly modified except through the execution of actions.

A process is composed of a series of actions. The types of process actions in the governance model are the following.

- A *notification action* alerts the client of some task that needs to be done, and can contain detailed instructions. Concretely, the notification arrives by email or phone, and the action might be the renewal of an insurance contract, a reminder to empty the cash register at the end of the day or instructions on how to patch a CMS application.
- A *confirmation action* is what a client uses to confirm in a provided Web-form that that he has undertaken a certain manual task. Execution of this task is used for attestation, i.e., to demonstrate whether the client has executed his attributed tasks.
- An *inform action* is used by a client to communicate with some other user. For instance, this could be used by a manager informing employees about the Internet usage policy. In this way, an employee cannot later claim unawareness in the event of a dispute.
- A *burden of proof action* **BoP** permits a client to provide specific proof of the implementation of some task for attestation. For instance, if the task is the installation of anti-virus software on his PC, the proof might be the screenshot of the folder with the software or the list of running applications containing the anti-virus software.

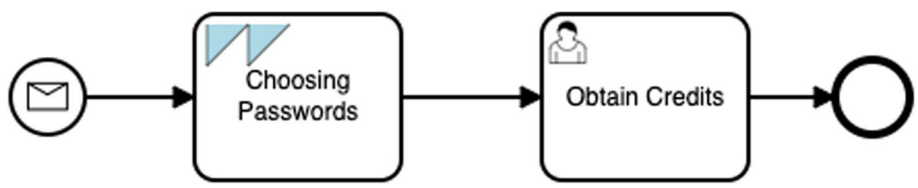

Fig. 7 Micro-learning process can be triggered by a process managing password leaks

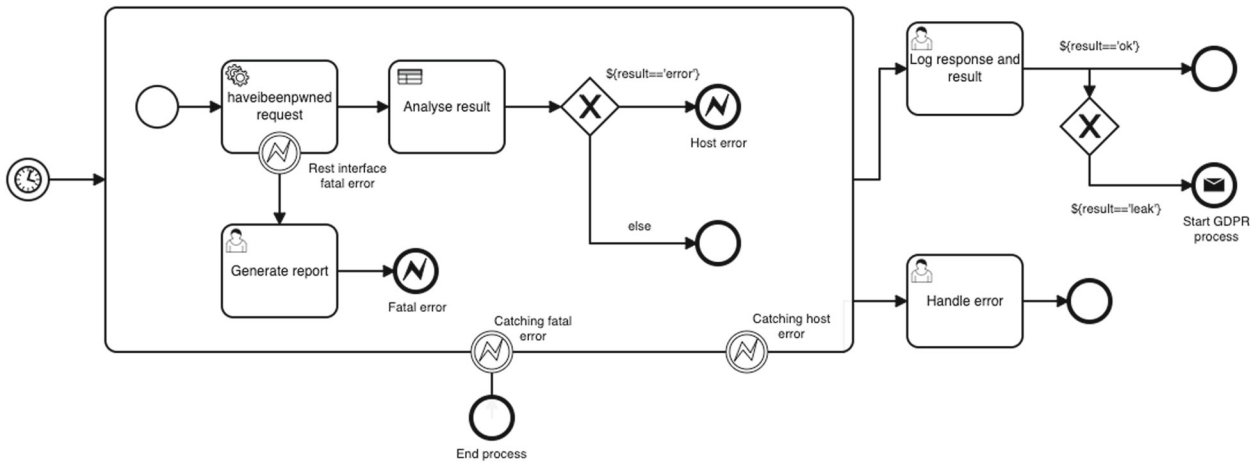

Fig. 8 Process watching for Data Leaks

- A *custom action* is any other type of action that the provider may care to implement. One example is a service task that scrapes Web sites for indications of a data leak involving the company (e.g., via *haveibeenpwned.com*). Another example of a custom action is an on-line training course.
- An *expression action* updates the dashboard's client variables, e.g., for a process accompanying the round of a security guard, the expression actions updates the time of the guard's last check.

As mentioned, one objective of the model is to permit *attestation* of the implementation of governance processes by a auditor. Two classes of information are available for this purpose. First, the *status* of process execution. This consists of the list and times of processes executed, as well as the state (current action and previous execution times) of currently executing processes. Second, data uploaded through BoP actions also constitutes attestation evidence.

An overview of our architecture is presented in Fig. 4. Governance security processes run on a process server on the cloud. The service can be used simultaneously by different clients, so there is a user management database. The service provider may include other useful services such as a CVE (Common Vulnerabilities and Exposures) feed that is used to trigger client processes when a relevant virus alert is detected. As will be explained in "Implementation" section, programs called compliance agents (CA) are placed within the client IT infrastructure. These are responsible for uploading data to server processes through BoP actions.

Governance processes

Processes are programmed using the *Business Process Model and Notation* (BPMN) standard from the Object Management Group [18]. The language is now widely used in industry. The primary purpose of this language is a means of specification that is accessible to all stakeholders of a company, and not just the IT developers. This explains its largely shape-based, non textual, syntax. This argument is particularly relevant to security and compliance since the specification of security policies and their enforcement depend on end-users, IT administrators, management, legal and HR.

BPMN specifications are not just about modeling. Many platforms exist to execute BPMN processes. Open-source BPMN platforms include Bonita BPM [4], Red Hat's JBoss BPM suite jBPM[1] and Camunda[2].

The tasks of the BPMN processes do the actual work. Each BPMN platform has its own types of implementation tasks, though they are generally similar. The platform used for our implementation is Camunda, and has the following types of tasks, illustrated in the small process example of Fig. 5.

- The first task, "Email Alice", is a *message task*. This is used to implement the Inform action of the governance model.
- A *service task* is an automated work item task. The Camunda platform allows Java or Groovy code to be linked to this task and executed when the task is activated. Our implementation uses this task type to implement notification and custom actions. An example custom action that we developed is a training task that presents video content to users. Expression tasks are also implemented as service tasks.
- The third task in the process of Fig. 5 is a *user task*. In our implementation, we use this to implement

[1] jbpm.org
[2] www.camunda.com

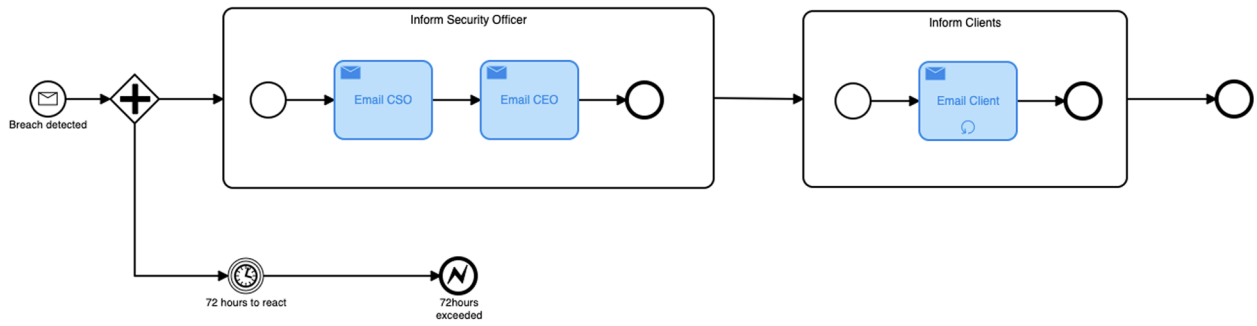

Fig. 9 Process triggered when data leak detected

confirm actions within a Web form. These forms are the means by which users may manually update variables. The task is executed by a user logging into the system, loading and submitting the Web form.

Another useful feature of BPMN is the swim-lane concept. This is a row that is drawn over tasks, and to which a user or group of users is assigned. In Fig. 5, user Bob is attributed the three tasks. The swim-lane denotes authorization and responsibility. The owning user or group is the only user allowed to execute tasks in the swim-lane. Swim-lanes are thus the means that access control is implemented, and in particular, how different tasks can be attributed to employees with different roles within a client company.

Returning to the example of Fig. 2, one of the asset-threat pairs identified was theft of the credit card reader. The proposed process is to store the item in the cabinet each night before leaving. The process deployed in our framework is shown in Fig. 6.

The start event of this process is a BPMN *timer event*. This event is programmed to start a process at a designated or at regular intervals. In this case, the process is scheduled each evening. The first task of the process is a notification task. This task type sends a message to the client. The message text and receiver identifier are specified as parameters to the task. The second task is a user task, and represents the action that the client is requested to do. The user task is a Web form with has explicatory text about what the client is requested to do, and a confirm button that the client uses to claim that the task has been done. This is the platform's manner of noting actions done on the client side.

Sample process portfolio

For a micro or small company, several risks are recurrent. One is loss or theft of passwords. Another is data leakage, where either they are the subject of the leakage, or their client data that they manage gets leaked. In the latter case, the dispositions of the GDPR come into play. Another risk is for micro-companies is a bad on-line review. Though this risk is not traditionally considered a security risk, all micro-companies insist upon this risk since a bad review can have a serious impact on their business. In any case, security is about protecting assets – and reputation, though intangible, is definitely a company asset.

In helping a client company, the governance provider selects process descriptions from a database of processes,

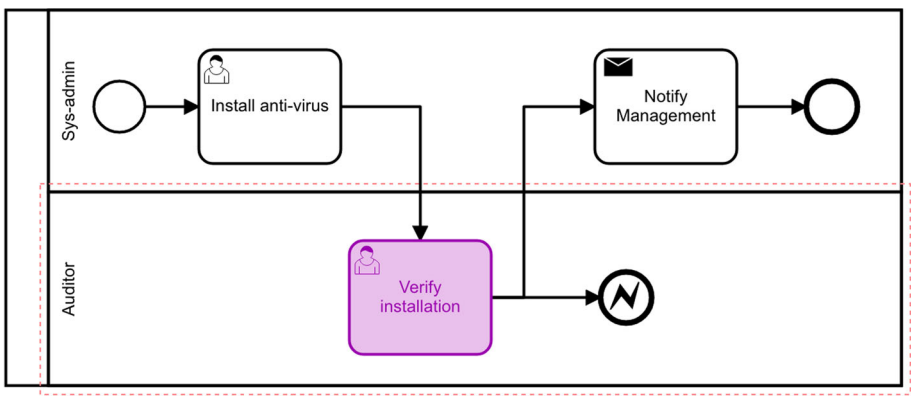

Fig. 10 Requirement for validating action has been performed

Fig. 11 Anti-virus installation process with BoP

and customizes them for the client. Given the aforementioned risks, one process deployed could be a micro-training process, c.f., Fig. 7. Note that this process has a *message* start event, meaning that the process can be started by another process that sends it a message. The governance provider may also start it manually on the client's behalf. This process has a custom training task that, in this case, displays a video about password management. This message event approach allows security training to be delivered at the most pertinent moment, e.g., by a process that detects a password leak sending a message to trigger the training.

A process that checks for data leaks is shown in Fig. 8. It contains a service task that integrates a call to the REST interface of the *haveibeenpwned.com* site. The addresses to check for are stored as process variables. In the event of a data leak detection, the end event is a message event that triggers the GDPR process shown in Fig. 9.

The GDPR process takes dispositions required under the GDPR. The process has a parallel treatment. In the top part, a sub-process has tasks to inform members of the company about the leak, followed by a sub-process to inform clients. In the bottom part, a timer is triggered which runs for 72 h – the limit within which clients must be informed of the leak.

Other processes developed include a *reputation check* process that is triggered by a timer event. Currently, this contains a service call to the Google Places API to check for any negative reviews. Obviously, calls to any sentiment analysis site could be integrated. Another class of processes that we are working on are for data backup and restoration. While our server runs processes to explain how backups and restorations can be done, and which inform the client to trigger a backup, the actual backup and restoration are done using scripts within the client IT infrastructure.

Burdens of proof

Recall that a primary goal of our framework is to help demonstrate that processes are being implemented, and consequently security is being enforced. One reason for outsourcing process execution to a trusted third party is that it maintains a trace of process executions. For instance, when the confirm action is executed in the second task of Fig. 6, the server notes that the client handled the cabinet lockup request and notes the time. If ever a theft is signaled, the client can use this process execution history to demonstrate the actions taken prior to the theft.

However, sometimes stronger proofs than client confirmation are required to ensure that an action has been performed by the client. The scenario to implement is demonstrated using BPMN in Fig. 10 using the swimlanes. The system administrator installs anti-virus software. The next task executed is the responsibility of the auditor whose role is to validate the installation. Only then can the system administrator execute the next step of the process – informing management in this example.

Our goal is to automate the role of the auditor. To this end, we extend the BPMN model with annotations called **burdens of proof** (BoP). These are a form of intermediate event within a process. A BoP acts as a synchronous rendezvous point; process execution can only continue

Antivirus

U	Input +	Output +
	Software count count	OK
	integer	boolean
1	${count==0}	false
2	${count>0}	true

Fig. 12 DMN Decision table for antivirus installation process

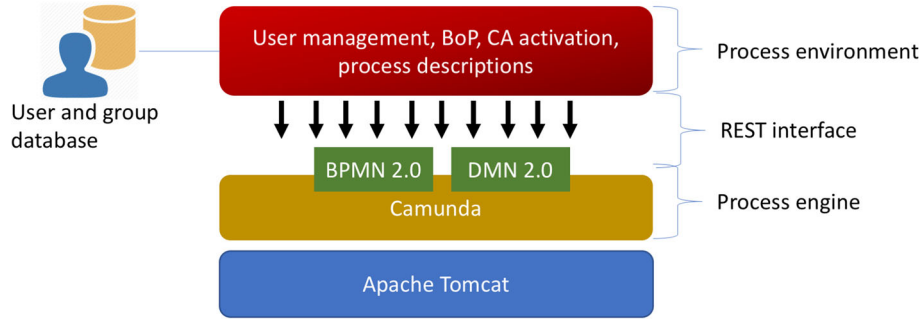

Fig. 13 Process Platform Stack Architecture

once evidence has been provided to a burden of proof and then validated. Figure 11 shows a simple example with a burden of proof element. A client installs an anti-virus software and then informs management. Before doing so, a proof of installation must be sent to the server. Without this, process execution does not continue to the notification task.

The service provider and company define a *policy* for a BoP that defines the evidence that must be furnished to the BoP. In our current implementation, this policy is expressed using *Decision Model Notation* (DMN) [12], and an example is shown in Fig. 12. DMN is a notation for decision modeling from the Object Management Group. Each decision is represented by a table, with input arguments and corresponding outputs. The inputs are taken from a process' variables at runtime.

The simplified example policy in Fig. 12 specifies evidence needed for demonstrating that an anti-virus was installed. The policy is based on a variable *count* that represents the number of recognized anti-virus applications installed on the client site. This variable is input to the policy object's decision table. Policies can have any number of input columns and there is a single output column to indicate whether the process can continue or not. In this policy, the BoP allows execution to continue so long as there is at least one recognized anti-virus application. The syntax of the cell expressions is defined by the tools we use in our implementation[3].

Implementation

We use Camunda's community edition[4] as the process engine of our platform. As shown in Fig. 13, Camunda supports BPMN 2.0 and DMN 1.1. Camunda is an open-source platform and is one of the leading BPMN platform providers around. The platform has a REST interface for manipulating process descriptions, process instances, variables and users. Camunda is packaged as a servlet and we run it over an Apache Tomcat Web server container.

For the process platform, the choice was made to use customizable off-the-shelf components. Our process environment runs in parallel to the Camunda/Apache server over a MERN stack (MongoDB, Express, React and Node). This environment manages the process descriptions, c.f., Fig. 14, with a link to an editor for creating and editing descriptions, a deploy and un-deploy function, as well as a start process button. Process deployment and starting are implemented using Camunda's REST interface for process control. In addition, the portal has a window with the list of currently running processes and active tasks that a user has to execute. Another window offers metrics on process descriptions and instances.

The portal's process editor, c.f., Fig. 15, is an extension of Camunda'a open source editor[5]. The extension includes support for burdens of proofs and the customized task types – training task, notification task, etc.

As mentioned, **compliance agents** are deployed on the side of the client organization. We currently run them as protected Ruby scripts, but since they communicate over HTTP, any technology can be used to implement them. The platform runs an *agent daemon* that receives HTTP calls from compliance agents and forwards them to the pertinent BoP element to trigger evaluation.

A compliance agent used for the anti-virus example is shown in Fig. 16. It is just a Ruby script that verifies the list of installed programs and compares to the recognized list of programs – McAfee and BitDefender in this example and submits the resulting *count* value. On Mac OS platforms, the list of installed programs is given in the */Library/Receipts/InstallHistory.plist* file.

Though we have extended BPMN with BoP elements, this extension is actually implemented using other BPMN elements. That is, a process from our framework that contains BoP elements is rewritten in pure BPMN as

[3] We are using the Camunda platform (www.camunda.com) which supports DMN. This implementation supports JUEL (Java Unified Expression Language) and FEEL (Friendly Enough Expression Language) expressions in cells.

[4] www.camunda.com

[5] https://github.com/bpmn-io

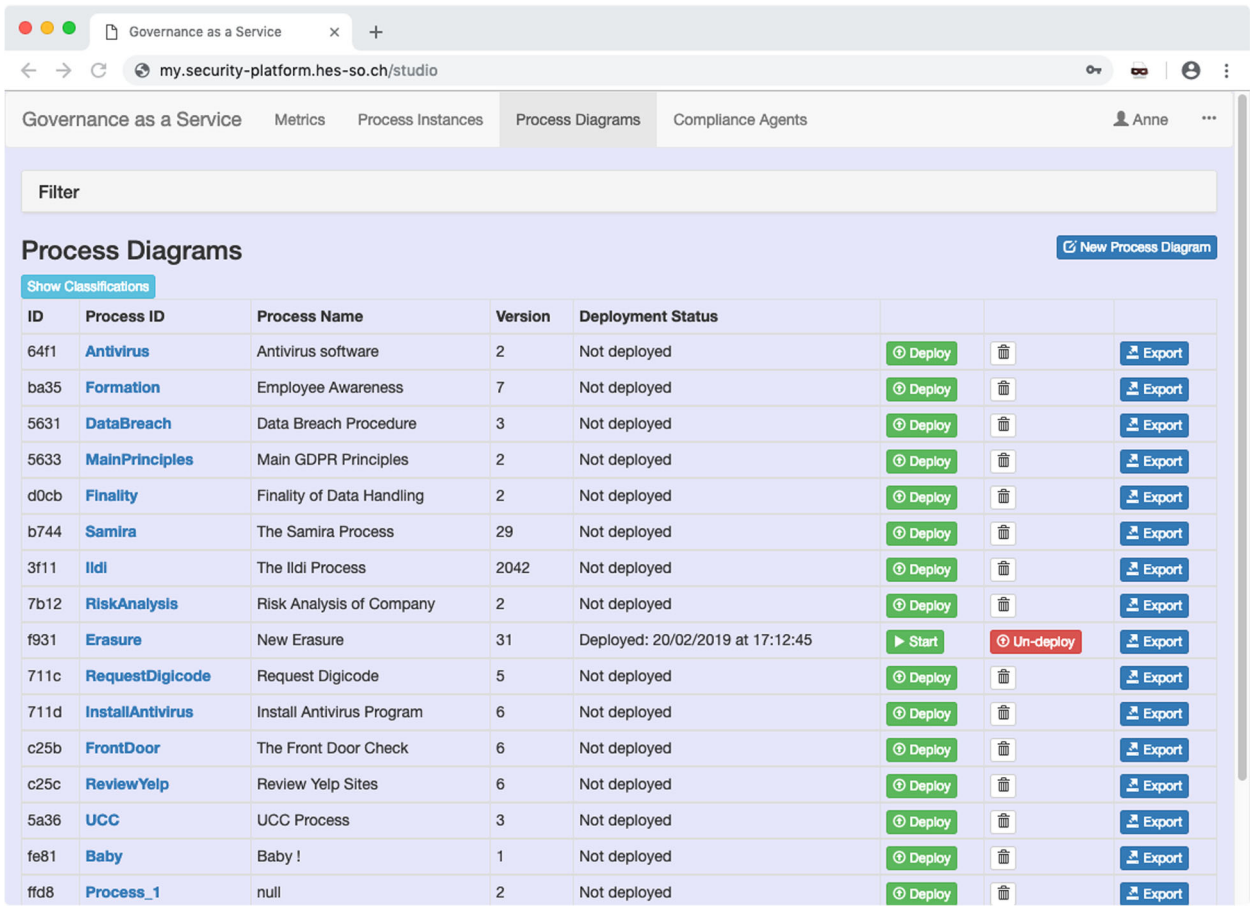

Fig. 14 Screenshot of process descriptions

soon as it gets deployed. This approach has the significant advantage that we did not need to modify the Camunda process engine, and that processes from our library can run on any BPMN 2.0 platform.

In the BPMN process transformation that occurs on deployment, each BoP is transformed into the *sub-process* task shown in Fig. 17. One of the task types supported by Camunda is a *Business Rule Task*. This task evaluates a DMN decision table. As mentioned, each BoP has an argument DMN policy decision table and this is what gets linked to the business rule task. The policy outputs true or false, depending on the evidence provided. A negative decision leads to the BoP generating an error event which is caught by the second sub-process in the Figure. Finally, the BoP gets triggered by a receive message event. This is triggered by a message sent to the platform, which is triggered by the HTTP request made by the corresponding compliance agent on the client IT system.

The compliance agent must be protected from tampering so that evidence is not fabricated. Currently, we run them as *setuid* enabled programs on our POSIX platforms, where the user identity attributed to the programs is that of an administrator of the process platform. The only drawback is this does not handle the threat of a malicious IT administrator. This is the subject of current work. Among the possible solutions, the Intel SGX (Software Guard Extensions) mechanism allows user code and data to be placed within memory *enclaves* [3]. Enclave memory cannot be directly accessed by code running in kernel mode. This means that we can run code that is safe from manipulation by kernel-level code, and thus, safe from manipulation by malicious administrators. An example of its use for secure databases is presented in [15].

Related work

BPMN has been used on several occasions for modeling security in the context of business processes [8]. The motivation for this is the undoubted success that the language has as a modeling and orchestration tool within industry, but since the language only specifies functional requirements, security needs somehow to be added to the process

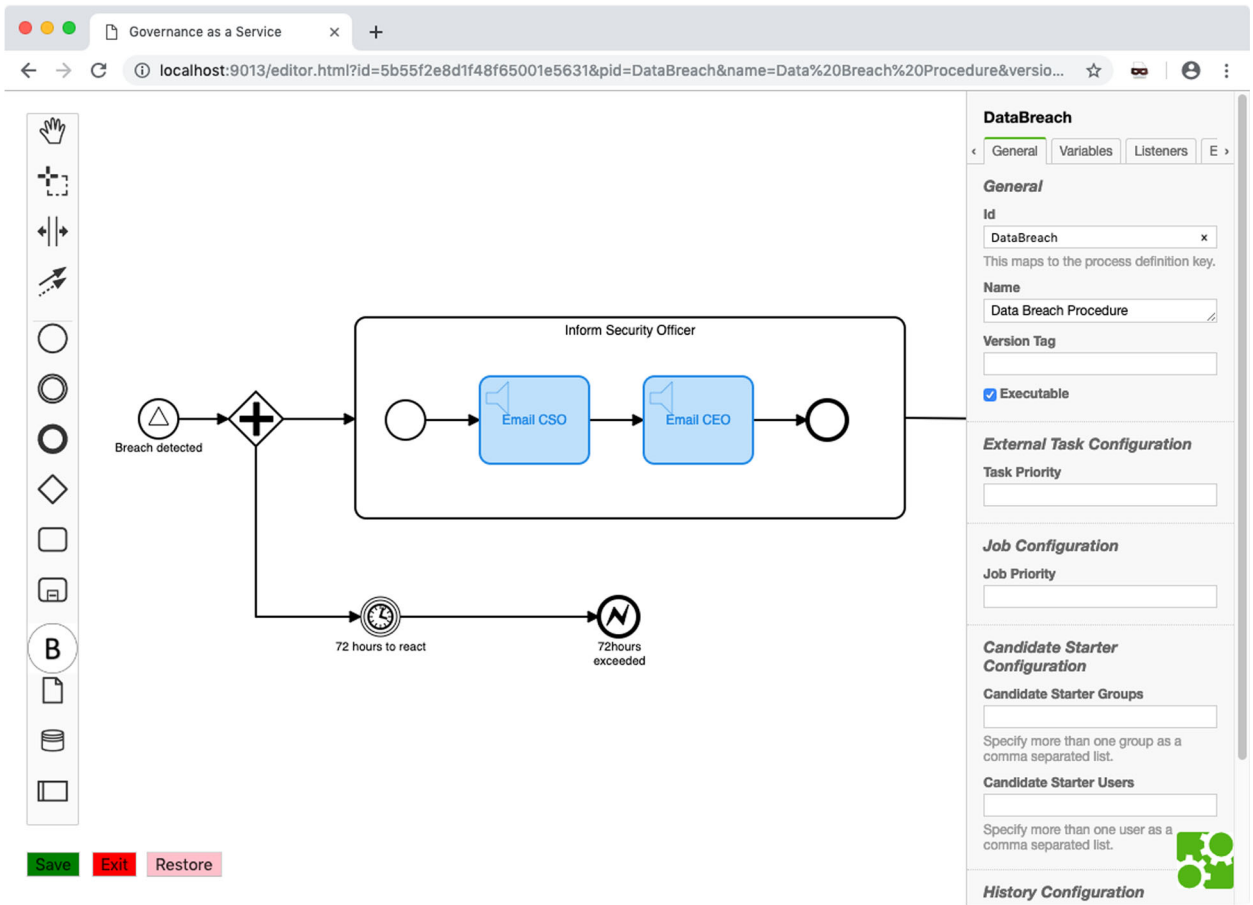

Fig. 15 Screenshot of process editor

descriptions. The general approach is to extend BPMN with security features.

Two examples of this are SecBPMN [17] and SecureBPMN [9]. SecBPMN extends BPMN with annotations for accountability, audit, authenticity, availability, confidentiality, integrity, non-repudiation and privacy. The result is a modeling language for secure business processes. In addition, the authors also develop an icon based query language in [17] that is used to verify security properties of a process. A key disadvantage of extending BPMN in this way is that compatibility with BPMN platforms is lost. SecureBPMN is an earlier effort, with fewer annotations. In our work, though we have added a new BPMN element – the BoP – there is a rewriting rule to transform this to pure BPMN. Further, our concern is not to add security to existing BPMN workflows, but to use BPMN to implement many of the security processes that companies need to implement.

In [1], the authors do present BPMN patterns for GDPR, e.g., consent to use data, right to access data, right of portability, right to withdraw, right to rectify, and right to be forgotten. These are extremely useful; in our work, we seek to implement a larger portfolio of security processes.

Another axe of related work is the use of cloud services for security. The *Cloud Security Alliance* (CSA) provide a definition of Security as a Service (SecaaS): a cloud infrastructure that provides security services such as identity management, data loss prevention, Web and mail security, Business Continuity and Disaster Recovery, as well as security information and event management. This underlines the trend that outsourcing security to third-party providers is now a viable option for companies. Nevertheless, the introduction of a new third-party has an implication on risk, as companies require that the cloud service be secure. In [19], the authors present a security model for securing cloud services. All data objects stored by the cloud server have annotations that can be used to express privacy and access control properties. These permit services to be constructed in a security-by-design approach. Such an approach could be applied to this paper's service.

Yet confiding security and data to a cloud service can be perceived as risky and many seek better transparency

```
# Compliance Agent for Anti-virus installation
#

antivirus_list = ['bitdefender virus scanner', 'macaffee']

total_count = 0
antivirus_list.each do |el|
  match = File.open('/Library/Receipts/InstallHistory.plist').grep(/#{el}/i)
  total_count += match.count
end

# Contact GOVaaS
uri = URI.parse("http://my.security.ch/complianceagent/execution")

header = {'Content-Type': 'application/json'}
user = {messageName: 'antivirusinstall',
        processVariables: {
          count: {
            value: total_count,
            type: 'Integer'
          }
        }
       }

# Create the HTTP objects
http = Net::HTTP.new(uri.host, uri.port)
request = Net::HTTP::Post.new(uri.request_uri, header)
request.body = user.to_json

# Send the request
response = http.request(request)
```

Fig. 16 Compliance agent for antivirus example

between cloud customers and providers [13]. Notably, while security techniques have improved, techniques aiming at security transparency and mutual accountability have lagged behind. In [11], a model for formalizing the trust relationship with the cloud service is presented. This is important since local IT infrastructures delegate control and pass sensitive data to the service. We could leverage this model for passing burden of proof documents to the process server in future work.

The cloud is often seen as the ideal host for security operation centers [10]. However, these are designed with larger companies in mind. Small companies are often not economically attractive. As a result, they adopt generic solutions like standard anti-virus software and firewalls installed by their telephony provider. Obviously these are important solutions, but security also requires process that are customized for each company. We contend that the high degree of process automation possible and the overlap of risks between companies create the economy of scale needed to make the service a viable one.

Adamant is another framework for security automation [2]. The goal is to be aligned to the ISO 27001 standard. The UML language is used for describing assets in a catalog. A state machine model then describes asset states, e.g., for compliance, and the Spring expression language is used for this. States evolve as actions are reported to the system. Our work complements Adamant by focusing more on the operational processes that must be put in place for governance.

Conclusions

This paper has presented the design and implementation of a model for security governance. Increasingly, it is necessary for companies to demonstrate implementation of security processes so that they do not become liable in the event of an attack leading to a data loss, and so that they can demonstrate compliance with legislation like the GDPR. These requirements led us to a cloud solution where the participation of a trusted independent party is leveraged for its competence in the security domain as well as for proof of process implementation.

The service uses the BPMN language to express security and compliance processes. These are executed in real-time on a process engine on the cloud. BPMN is extended

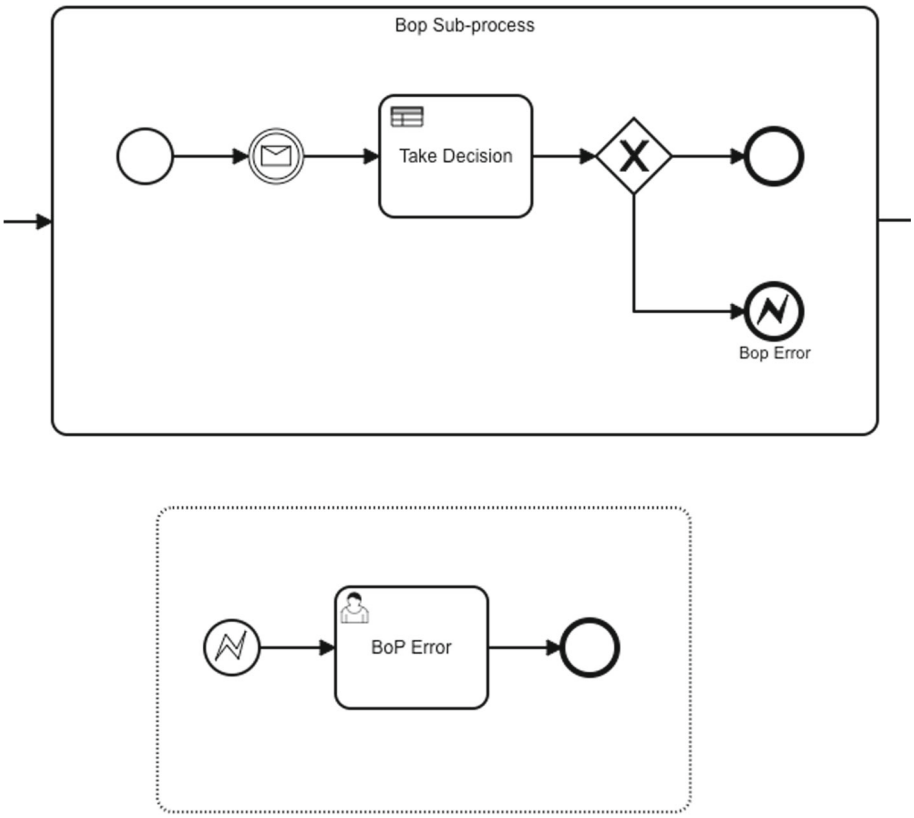

Fig. 17 Implementation of the BoP element

with annotations for burdens of proof. These represent points in the process where evidence must be furnished and validated for execution to continue. These constitute proof of implementation of the process for management and auditors. The BoP elements are implemented in BPMN, so processes from the library can be run on any BPMN platform.

Future work includes investigating implementation techniques for compliance agents that protect their execution in environments where the administrator is hostile.

Abbreviations

BoP: A *Burden of Proof* is an element that we added to BPMN to model the exchange of proof between a client and the cloud service; BPMN: The *Business Process Model and Notation* is a modeling language defined by the Object Management Group (OMG) for expressing business processes and workflows; CA: A *compliance agent* is a program placed on the client site that exchanges proof data with the process executing on the cloud service. DMN: The Object Management Group (OMG) have defined the *Decision Model and Notation* standard for formally modeling decision trees; GDPR: The *General Data Protection Regulation* is an EU regulation that defines conditions on the handling of personal data in IT systems. The GDPR came into effect on May 25th 2018; ISO/IEC: The *International Organization for Standardization* (ISO) and the *International Electrotechnical Commission* (IEC) have defined a family of standards for information security

Acknowledgements

The author is grateful to the RCSO ISNET research program for funding this research. The author thanks Sylvain Muller for his help developing the platform.

Authors' contributions

The author read and approved the final manuscript.

Authors' information

The author is an associate professor at the Geneva School of Business Administration at the University of Applied Sciences and Arts of Western Switzerland. Research at this university is generally applied, and conducted in close collaboration with local industry.

Funding

This research has been funded by a research program of the University of Applied Sciences and Arts of Western Switzerland named RCSO ISNET, grant number 86112/ES-ISNET18-04.

Availability of data and materials

The model proposed in this paper is inspired by the author's experience in industry and on informal discussions with companies. There is no formal data set. Any further information is available on demand from the author.

Competing interests

There are no competing interests involving the author in the execution of this research work.

Received: 23 April 2019 Accepted: 10 December 2019
Published online: 26 December 2019

References

1. Agostinelli S, Maggi FM, Marrella A, Sapio F (2019) Achieving GDPR compliance of BPMN process models. In: Information Systems Engineering in Responsible Information Systems - CAiSE Forum 2019, Rome, Italy, June 3-7, 2019, Proceedings. pp 10–22. https://doi.org/10.1007/978-3-030-21297-1_2

2. Brunner M, Sillaber C, Breu R (2017) Towards automation in information security management systems. In: 2017 IEEE International Conference on Software Quality, Reliability and Security, QRS 2017, Prague, Czech Republic, July 25-29, 2017. pp 160–167. https://doi.org/10.1109/qrs.2017.26
3. Cedric Xing B, Shanahan M, Leslie-Hurd R (2016). IntelÂ® software guard extensions (intelÂ® sgx) software support for dynamic memory allocation inside an enclave 06:1–9
4. Chabanoles N, Ozil P, Farrance M (2015) Bonita BPM: an innovative bpm-based application development platform to build engaging, user-oriented business applications. In: BPM (Demos). pp 21–24. http://ceur-ws.org/Vol-1418/paper5.pdf
5. Gikas C (2010) A general comparison of fisma, hipaa, ISO 27000 and PCI-DSS standards. Inf Secur J A Glob Perspect 19(3):132–141
6. Government OpenAccess (2019) SME Owners Still in the Dark about GDPR. www.openaccessgovernment.org/sme-owners-gdpr/57656
7. Horkoff J, Jeusfeld MA, Ralyté J, Karagiannis D (2018) Enterprise modeling for business agility. Bus Inf Syst Eng 60(1):1–2
8. Lins FAA, Sousa ETG, Rosa NS (2018) A survey on automation of security requirements in service-based business processes. Int J Web Eng Technol 13(1):3–29
9. Mendling J, Weidlich M (eds) (2012) Business Process Model and Notation - 4th International Workshop, BPMN 2012, Vienna, Austria, September 12-13, 2012. Proceedings, volume 125 of Lecture Notes in Business Information Processing. Springer
10. Miloslavskaya NG (2016) Security operations centers for information security incident management. In: 4th IEEE International Conference on Future Internet of Things and Cloud, FiCloud 2016, Vienna, Austria, August 22-24, 2016. pp 131–136. https://doi.org/10.1109/ficloud.2016.26
11. Mont MC, Matteucci I, Petrocchi M, Sbodio ML (2015) Towards safer information sharing in the cloud. Int J Inf Sec 14(4):319–334
12. Object Management Group (OMG) (2019) Decision Model and Notation (DMN). OMG Document Number formal/dtc/18-06-04. https://www.omg.org/spec/DMN/About-DMN/
13. Ouedraogo M, Mignon S, Cholez H, Furnell S, Dubois E (2015) Security transparency: the next frontier for security research in the cloud. J Cloud Comput 4:12
14. Picahaco AM, Mesquida AL, Alcover EA, Fluxà B (2010) ISO/IEC 15504 best practices to facilitate ISO/IEC 27000 implementation. https://doi.org/10.5220/0003001001920198
15. Priebe C, Vaswani K, Costa M (2017) Enclavedb: A secure database using sgx. In: 2018 IEEE Symposium on Security and Privacy, SP 2018, San Jose, CA, USA, May 22-26, 2018. pp 3–18. https://doi.org/10.1109/sp.2018.00025
16. Saleem J, Adebisi B, Ande R, Hammoudeh M (2017) A state of the art survey - impact of cyber attacks on sme's. In: Proceedings of the International Conference on Future Networks and Distributed Systems, ICFNDS 2017, Cambridge, United Kingdom, July 19-20, 2017. p 52. https://doi.org/10.1145/3102304.3109812
17. Salnitri M, Dalpiaz F, Giorgini P (2017) Designing secure business processes with secbpmn. Softw Syst Model 16(3):737–757
18. Schleicher D, Fehling C, Grohe S, Leymann F, Nowak A, Schneider P, Schumm D (2011) Compliance domains: A means to model data-restrictions in cloud environments. In: Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2011, Helsinki, Finland, August 29 - September 2, 2011. pp 257–266. https://doi.org/10.1109/edoc.2011.22
19. Verginadis Y, Michalas A, Gouvas P, Schiefer G, Hübsch G, Paraskakis I (2017) Password: A holistic data privacy and security by design framework for cloud services. J Grid Comput 15(2):219–234

Publisher's Note