Hybrid Security Approach for Behavioural Privacy of Business Processes in a Cloud Environment

Farah Abdmeziem[1(✉)], Saida Boukhedouma[1], and Mourad Chabane Oussalah[2]

[1] LSI Laboratory, University of Science and Technology Houari Boumediene, Algiers, Algeria
{fabdmeziem,sboukhedouma}@usthb.dz
[2] LS2N Laboratory, University of Nantes, Nantes, France
Mourad.oussalah@univ-nantes.fr

Abstract. Business processes are surely at the heart of companies but they are also the target of numerous security attacks, especially when using nowadays technologies like Cloud computing. Hence, ensuring business processes' security is one of the main concerns in both industry and research area. Securing a business process involves securing its three main aspects namely, the informational, the logical (behavioural), and the organisational aspect. While most of the works in the literature focus on the security of the informational aspect, Goettelmann et al. in [9] propose an obfuscation technique that aims to guarantee explicitly the security of the logical aspect of a business process i.e. persevering the privacy of the company's know-how expressed through the business process. This paper proposes an alternative solution to the obfuscation technique that is based on the strengths of this latter while addressing its identified limits. Our approach combines the ideas of the obfuscation technique and security annotations. The proposed solution is described via a metamodel that exhibits the main concepts it is based on, and a global functional architecture describing the principal steps of the solution, mainly a deployment configuration of the business process, meeting a set of defined constraints (and rules). A comparison between the obfuscation technique and our proposed approach according to security level and deployment costs is also illustrated, through examples.

Keywords: Business process • Security • Multi-Cloud environment • Obfuscation • Security annotation • Know-how privacy • Constraints • Process fragment

1 Introduction

A business process (BP) is a set of tasks (activities) that are to be performed in a logical manner by actors (human or system) using and producing resources (data and artefacts), so as to fulfil an overall business objective. A business

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14104, pp. 271–289, 2023.
https://doi.org/10.1007/978-3-031-37105-9_19

process is defined through three main aspects namely the informational aspect which pertains to the data used and generated throughout the life-cycle of the business process; the logical aspect that encapsulates the company's know-how and expertise; and finally the organisational aspect which concerns the allocation of tasks to stakeholders within the company. Given that business processes are essential to a company's operations, they are often targeted by security attacks. This is especially true now that companies are increasingly leveraging new technologies such as cloud computing to build and deploy their business processes, which introduce new security risks. In this vein, in our previous work [8], we studied business processes security and we gave an overview of security mechanisms used in the literature. During our study we found out that most of research works focus on the security of the informational view [1,6,7,18] and to the best of our knowledge, the work of Goettelmann et al. in [9] is the first to focus solely on securing the logical aspect of business processes, by proposing an obfuscation technique. The present work is founded on the strengths of the obfuscation technique while seeking to address its limits. We propose an hybrid security approach by first proposing a multi-level security annotation step that consists in annotating business processes tasks with four labels namely Highly-sensitive, Sensitive, Complementary or non-sensitive according to the sensitivity degree of each task. Then, we describe a global functional architecture that exhibits all steps of the proposed solution. Specifically, we describe an algorithm to build a deployment configuration (meeting a set of defined constraints) of the business process in a set of available Clouds (multi-Cloud environment). Notice that the set of constraints (and rules) used for building a deployment configuration is checked for consistency using a SAT solver, in order to avoid constraints violation. The principal concepts of our solution are also described through a meta-model that globally outlines the links between these concepts. The remain of this paper is structured as follows: Sect. 2 gives an overview of research works proposed to ensure business processes security. In Sect. 3, we detail the obfuscation technique proposed in [9] and we highlight its strengths and limits. To palliate this limits we present, first, in Sect. 4 our meta-model that represents the core concepts of our hybrid security solution. Followed by the details of this latter in Sect. 5. Finally, we illustrate our solution on a motivating example in Sect. 6 before concluding and outlining future work directions in Sect. 7.

2 State of the Art

In this section, we give a glimpse of the approaches tackling business processes security. We categorize the proposed solutions according to the business process aspect tackled by the approach especially, the logical and the informational (data) aspects.

Security of the Informational Aspect: In [2], Altuhhova et al. propose an extension to the Business Process Modelling Language (BPMN) to take into consideration security concerns. The extension is based on Dubois et al. work [5], which

propose a domain language that helps to extend different languages with the notion of risk management. Furthermore, Ahmed et al. in [1], propose a method to assign security requirements to business processes, the solution is based on risk-oriented patterns. Jensen et al. propose in [12], through three architectural patterns to achieve security in a cloud environment by using multiple distinct clouds (Multi-cloud environment). One of the patterns consist in *Partition of application system into tiers* and aims to protect from data leakage by separating the logic from data. Turki et al. propose in [20] to enrich BPMN with security notations described as padlocks. Different padlocks are proposed to match security requirements as confidentiality, integrity and, privacy. In the same vein, Rodriguez et al. in [19] also propose graphical annotations in the form of padlocks with specific codes to specify the security requirements, for example, the code CO refers to confidentiality. Another interesting solution is proposed by Watson in [22], it is based on the Bell-LaPadula Multi-level security model [4], where subjects represent the business process services and objects the data in input and output to/from the services. At last, we have the work in [10] where Goettelmann et al. propose an approach to ensure sensitive data exchange of business processes in a Cloud environment. The solution consists in transforming a centralized orchestration into a choreography under security constraints. Moreover, In [11], Han et al. propose an optimal distribution of business processes using a cloud and user-end resources, with the aim to protect sensitive data. In the same line, Pavoa et al. in [18] propose an approach to decompose business processes for cloud deployment, while preserving data safety constraints, monetary costs and performances requirements. Furthermore, the authors of [6], propose an automatic decomposition of a business process for its deployment on premises and on a Cloud resources to preserve sensitive data. At last, in [7] an approach that consists in obfuscating sensitive data before its storage in the Cloud services is proposed. The solution is based on ASCII characters to obfuscate data and reduce its size.

Security of the Logical Aspect: To protect the logical aspect, Jensen et al. in [12], propose through another architectural pattern, to split the logic into fine-grained fragments that could be deployed in distinct clouds. However, the solution does not provide specific details on how the splitting process could be carried out. On similar note, Gottelmann et al. in [9], propose a semi-automatic solution that consists in splitting the business process logic into fragments that should be deployed into distinct clouds. The fragmentation is based on the AND split and X(OR) gateways of BPMN[1]. We have built our work around this solution, and in Sect. 3, we provide further details about it. Nacer et al. in [17], go one step further and tackle the issue of malicious clouds collusion in case of deploying a business process into a multi-cloud environment. Indeed, splitting a business process into different fragments can surely preserve the know-how of companies but can not prevent from an eventual coalition between the involved clouds, to trace back the business logic. For that, Nacer et al. propose to add fake

[1] Business Process Modelling Notation.

fragment to the business process into strategic points, this can result in increased complexity of the business process and may require the involvement of multiple clouds, making it more challenging to orchestrate a collusion between them. In the same vein, Nacer et al. propose in [16] a metric for evaluating the privacy level of the business process logic because they claim that even though adding fake fragments can reduce cloud collusion but a risk persists and it is important to be able to measure it. Such a metric is important for cloud clients who want to compare different deployments solutions. However, adding fake fragments could result in increasing the deployment costs of the business process. Hence, in their further work Nacer et al. in [15] consider this issue and propose a solution to find a balance between costs and security risks of the logical aspect of a business process deployed in a multi-cloud environment.

3 The Obfuscation Technique

In this section, we give more details about the obfuscation technique proposed in [9] to preserve the companies know-how. After analysing a number of business processes, the authors found that sensitive tasks of a business process are those tasks where decisions are taken and where synthesis are made. Furthermore, they established a link between decisions tasks and the opening X(OR) BPMN gateways (Fig. 2) and, between synthesis tasks and, the closing AND join BPMN gateways (Fig. 1). Finally, the authors found that generally, a decision task has a complementary fragment that comes after the closing X(OR) gateway (Fig. 2) and, the complementary task of a synthesis task is, the task or fragments that triggers an AND gateway (Fig. 1).

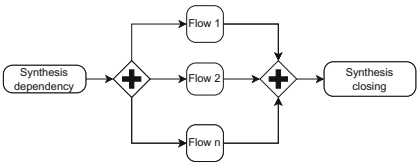

Fig. 1. Synthesis task

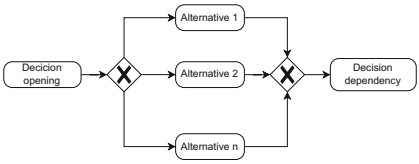

Fig. 2. Decision task

This analysis helps to automate the detection of sensitive task. For a secure deployment of a business process in a cloud environment, the authors proposed to split the business process into fragments and to deploy them into a multi-cloud environment. The solution is based on three main constraints: the trust-level, the co-locate and the separate constraints as follows: The co-locate and separate constraints state whether two fragments of a business process can be deployed in a same or, separate clouds respectively.

1. Separate the complementary fragments;
2. Separate the different alternatives of a X(OR) split gateway.
3. Deploy the synthesis and the decision fragments in clouds providing high trust level.

The solution is applicable only in the context of well-defined BPMN-based workflows and is based specifically, on the AND-join and X(or) split gateways whilst, the solution does not take into consideration the sequential pattern which is one of the most commonly used patterns in BPMN diagrams. Indeed, we could easily have the case of a BPMN diagram with only a sequence of sensitive tasks. Plus, while the proposed solution can provide satisfactory results in certain scenarios (e.g., when the fragments before an opening X(OR) gateway and after a closing AND gateway are highly sensitive), we contend that this is not always the case. To illustrate this point, consider the example depicted in Fig. 3, which outlines the process for creating a new customer bank account. After the evaluation of the client's supporting documents, the bank employee decides whether to accept or reject the request. In case of acceptance, an account number is allocated to the client and then, simultaneously, the service in charge of making credit card is notified and, a welcome pack for the client is prepared. At the end of the process, a notification is sent to inform the client if his/her request has been accepted or rejected. Following the proposed constraints in [9] we have come to the process partitioning depicted in Fig. 4 that shows the partitioning of the business process on three clouds. Notice the use of the tasks send/receive that allow the communication between the different fragments, which is a commonly used method in process choreographies, and the fictive task added in the exclusive OR gateway when one of the alternative is deployed on another Cloud. Hence, three clouds are needed for a secure deployment of this business process. Upon analysing the fragments or tasks that are identified as sensitive according to the Synthesis/Decision and Synthesis-dependency/decision-dependency constraints, we observe that the majority of these tasks are not actually sensitive, with the exception of the "allocate account number" task. Indeed, a bank may want to keep secret its strategy of allocating accounts to the client. Moreover, the tasks "Evaluate supporting document" and "Send final notification" are considered as dependent. However, in this case, deploying these two tasks on a same Cloud would not reveal a strategic and important information about the bank know-how. To conclude, the strengths and the drawbacks of the obfuscation technique could be summarized in the following items:

1. Strengths:
   - Automatic detection without human intervene;
   - The notion of complementary tasks is important as non-sensitive tasks once put together could reveal important know-how;
   - The separate and co-locate constraints that indicate how and where to deploy each fragment of the business process.
2. Drawbacks:
   - Relying only on the X(or) split and AND join gateways would not cover all the sensitive tasks of a business process. Hence, in some cases the fragmentation of the business process won't be efficient in term of security;

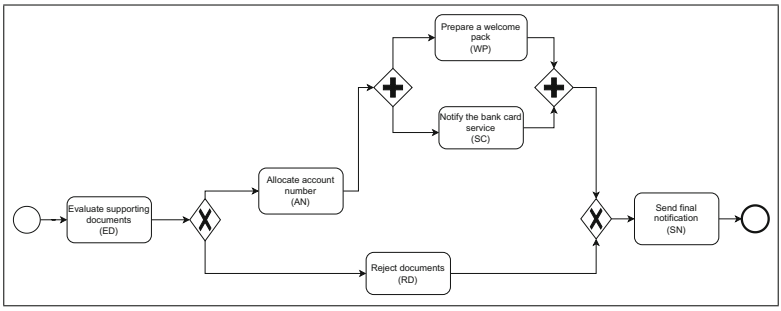

Fig. 3. BPMN process for Bank account creation

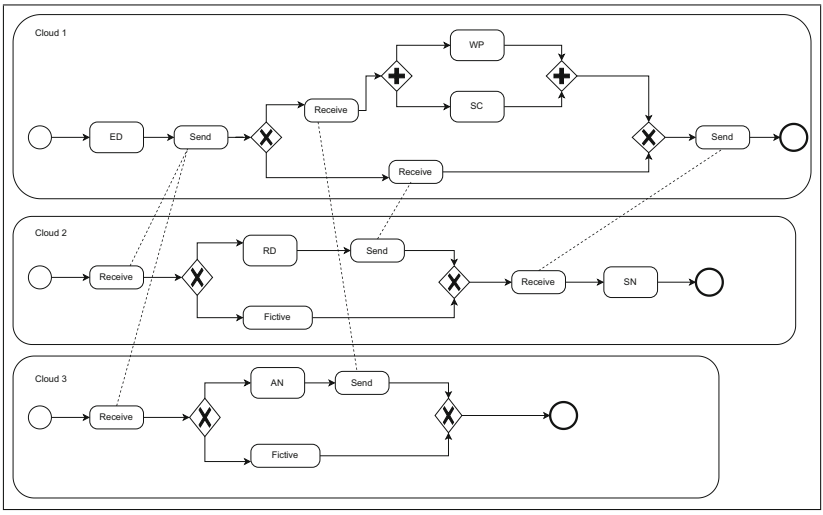

Fig. 4. Fragmentation of the bank account creation BP according to the obfuscation technique

– Conversely, sensitive tasks of a business process would not be necessarily the fragments before an opening X(OR) gateway and after a closing AND gateways;
– Same for the complementary task that could be at any location of the business process and not necessarily before/after X(OR) split and AND join gateways.

Consequently, these drawbacks can result in a fragmentation of business processes, which may not be effective in terms of security, or incurring significant costs due to the use of unnecessary multiple clouds in some cases.

In the next section, we propose an alternative solution to detect the sensitive tasks and, we enrich the proposed co-locate and separate rule and constraints, to ensure a secure deployment of business processes in a cloud environment.

4 Meta-model of Concepts: Hybrid Security Solution

Before digging into the details of our proposed approach, we introduce, first, a meta-model that serves as a conceptual foundation for our proposed security approach, which relies on the obfuscation and the annotation techniques. The meta-model, illustrated in Fig. 5, is designed to safeguard the know-how privacy of companies that are at risk of compromising their sensitive information by migrating to a cloud environment. Our hybrid approach leverages annotations to achieve this goal. Specifically, the know-how of a company is encapsulated in a business process, which can be modeled using BPMN (Business Process Modeling Notation) and is, by definition, composed of a set of tasks, and may contain X(or) /AND gateways to synchronize and link tasks. We distinguish two types of tasks namely business tasks, which encapsulate the business logic, and communication tasks, which permits the communication between different tasks. Business tasks can be annotated with sensitivity levels (i.e., highly-sensitive, sensitive, non-sensitive, or complementary). These annotations trigger security constraints "Separate," and security rules "Co-locate," between tasks. Applying these constraints and rules generates a deployable fragments of the business process. Finally, cost metrics could be used to evaluate the deployment cost of a business process in a multi-cloud environment using our hybrid security technique. This meta-model can be easily extended and instantiated to ensure the business process' behavioural privacy in a Cloud environment.

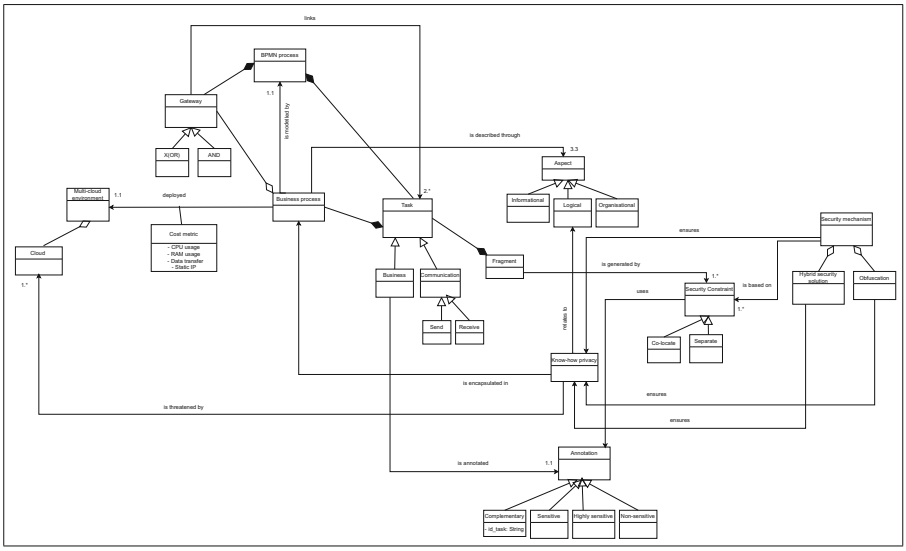

Fig. 5. Meta-model for the hybrid security solution

5 Hybrid Security Solution: Annotation and Obfuscation Based

In this section, we introduce our hybrid solution that combines security annotations and security constraints proposed in [9]. The primary aim of our solution is to secure the companies' confidential know-how encapsulated in business processes. The functional schema of our approach is depicted in Fig. 6 and comprises of four main phases: annotation, generation of constraints and rules, consistency check of the generated constraints and rules, and configuration generation. The output of each step of the process is represented with an ellipse form in Fig. 6. We provide further details on each phase in the following subsection.

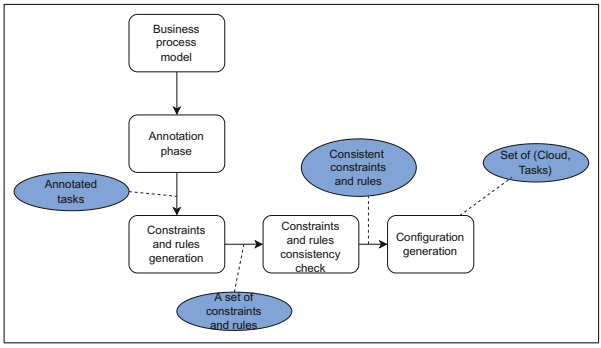

Fig. 6. Functional schema of the hybrid security solution

5.1 Annotation Phase

Annotations are a popular means of securing business processes, and they can be broadly classified into three categories: textual, graphical, and hybrid (a combination of textual and graphical). However, security annotations are often used to ensure business processes' data confidentially, integrity and availability [3,14,21]. In this work, we have managed to use textual security annotation to guarantee the privacy of the companies' know-how. Hence, a business analyst can use three annotations namely H.S: for highly-sensitive tasks, S: for Sensitive tasks, and CS for Complementary tasks to annotate the business process. In order to help the designer further in using these annotations, we give a brief definition of a highly-sensitive, sensitive and complementary tasks as follows: A **highly-sensitive** task is a task that encompasses strategic decisions or operations and at the same time, it uses or produces highly sensitive data.

A **sensitive** task is a task that either encompasses strategic decisions or uses/ produces sensitive data.

Finally, the **complementary tasks**: in line with the authors of [9], we also distinguish the complementary tasks that are tasks that could be non-sensitive

in their own but if deployed together in a sequential/alternative/parallel manner they could disclose confidential information about a company's knowledge or expertise. However, in our proposed solution, we do not associate these complementary tasks with specific AND/OR gateways; instead, they could occur at any stage of the workflow.

5.2 Fragmentation Constraints and Rules

After the annotation step, fragments of the business process are generated according to a set of constraints (Separate) and rules (Co-locate). The proposed constraints in [9] were adapted based on our proposed security annotations as follows:

1. Separate (H.S,H.S): deploy a pair of highly-sensitive tasks into different clouds;
2. Deploy H.S tasks into Clouds that satisfies the security level required by the highly sensitive task;
3. Separate (Tx,CS): deploy a task Tx and its complementary task CS into different clouds;
4. Separate(S,S): deploy a pair of sensitive tasks into different clouds;
5. Separate (H.S,S): deploy a pair of highly-sensitive and sensitive tasks into different clouds;
6. Co-locate(N.S,S): sensitive tasks and non-sensitive tasks can be deployed in the same Cloud;
7. Co-locate (N.S,S): highly-sensitive tasks and non-sensitive tasks can be deployed in the same Cloud;
8. Co-locate(N.S,N.S): non-sensitive tasks can be deployed in the same Cloud;

Constraint Consistency. After the constraints generation, it is important to check the consistency of these latter i.e. that a given constraint does not violate another one, for that we relied on a SAT solver to check the consistency of the conjunction of all the constraints. The SAT solver returns "SAT" i.e. there exits at least one configuration where tasks are allocated to Clouds while satisfying all the constraints and rules together, and "UNSAT" otherwise. For that, we encoded each constraint as a Boolean formula as follows:

(sensitive separate constraint) AND (co-locate constraint) AND (highly sensitive task separate constraint) AND (highly sensitive and sensitive tasks separate constraint) AND (separate complementary) AND (deploy highly sensitive in clouds with trust_level $\geq$ trust_level required).

Where as an indication, the sensitive separate constraint could be expressed as: (s1 AND NOT s2 AND c1 AND NOT c2) OR (NOT s1 AND s2 AND NOT c1 AND c2); Such as: s1: Sensitive task1, s2: Sensitive task2, c1: Cloud 1, c2: Cloud 2. Whereas, the Co-locate constraint could be expressed as: (ns1 AND s1 AND c1 AND NOT c2) AND ((ns2 AND c2) OR (ns2 AND NOT c1 AND s2 AND c2)); Such as: ns1: Non-sensitive task1, ns2: Non-sensitive task2.

5.3 Algorithm for the Configuration Generation

The Algorithm 1 is designed to return the final configuration, i.e., the set of tasks deployed in their corresponding clouds while ensuring the "Separate" constraints and "Co-locate" rule. To achieve this, the algorithm iterates over the set of business process tasks, which are sorted in descending order based on their sensitivity level. The highly-sensitive tasks, denoted by the H.S. annotation, are deployed first, followed by sensitive and complementary tasks. This deployment strategy is important to ensure that the clouds with the highest trust level are reserved for the highly-sensitive tasks. For each task in the set, the "TaskDeploymentScheme" procedure detailed in Algorithm 2 is called. This procedure checks the "Separate" constraint and returns a map of (tasks, cloud), i.e., where tasks should be deployed based on its sensitivity level. It is important to note that Algorithm 2 allows a task to be deployed in the first cloud that meets the security constraints stated above. However, there might be multiple clouds that meet these constraints without violating security. In these cases, a balance needs to be found between deployment costs and security constraints. While this is not considered in this work, it constitutes the object of one of our future research.

Table 1. Algorithm parameters notation

Parameters	
S	The set of the N available Clouds
C	The set of constraints on each task Ti
L	The minimum trust value required to deploy a highly sensitive task in a Cloud
B	The set of business process' tasks sorted in descending order
CloudByTask	A map to store the pairs (Task, Cloud) denoting that a given task is affected to a specific Cloud
isHsensitive	A function that checks if a task is annotated with the S annotation
isSensitive	A function that checks if a task is annotated with the H.S or S annotations respectively
Cid	ID of the complementary task of Tx

Algorithm 1. Final configuration

1: Input: N, C, S, L, B ▷ B is sorted in a descending order according to tasks' sensitivity level.
2: Output: Config
3: **for each** (task Tx in B) **do**
4: $Config \leftarrow TaskDeploymentScheme(Tx, N, C, S, L)$

Algorithm 2. Procedure: TaskDeploymentScheme

```
Input: Tx, N, C, S, L
Output: CloudByTask
bool ← true;
j ← 1;
if (isHSensitive(Tx)) then
    while (j <= N) do
        if (TrustLevel(Sj)>= L) then
            if (isEmpty(Sj)) then
                Deploy(Sj,Tx);
                CloudByTask.add(Tx,Sj);
                break;
            else
                for each (Ti in Sj) do
                    if (isHSensitive (Ti) or isSensitive (Ti) or equal(Ti.cid,Tx.id)) then
                        Separate(Ti,Tx);
                        j ← j + 1;
                        bool ← false;
                        break;
                if bool then
                    Deploy(Sj,Tx);
                    CloudByTask.add(Tx,Sj);
                    break;
        else
            j ← j + 1;
else
    if (isSensitive(Tx)) then
        while (j <= N) do
            if (isEmpty(Sj)) then
                Deploy(Sj,Tx);
                CloudByTask.add(Tx,Sj);
                break;
            else
                for each (Ti in Sj) do
                    if (isHSensitive (Ti) or isSensitive(Ti) or equal(Ti.cid,Tx.id)) then
                        Separate(Ti,Tx);
                        j ← j + 1;
                        bool ← false;
                        break;
                if bool then
                    Deploy(Sj,Tx);
                    CloudByTask.add(Tx,Sj);
                    break;
    else
        while (j <= N) do
            for each (Ti in Sj) do
                if (equal(Ti.cid,Tx.id)) then
                    j ← j + 1;
                    bool ← false;
                    break;
            if bool then
                Deploy(Sj,Tx);
                CloudByTask.add(Tx,Sj);
                break;
return CloudByTask
```

6 Use Case Scenario

Figure 7, depicts a BPMN process followed by a bank to grant or not a loan to a client, the process is inspired from [13]. When the client deposits his/her application, the process begins by retrieving the data needed from the application like the age and the salary of the client. Then, the installments are computed. With the retrieved data and the computed installments, the bank proceed the client's application according to its policy and, according to the results of this step the bank decides to reject or accept the client's application. The bank recognizes the potential gains that could be achieved by leveraging cloud computing environments to improve its operational efficiency. However, it is also concerned about the risk of disclosing its proprietary know-how to potential competitors or fraudulent cloud providers. Figure 8, depicts the Bank's business process annotated with our proposed security annotations. Indeed, the bank encapsulated the bulk of its know-how in the task "Process application" that computes sensitive data produced by the task "Compute installments". Hence, the "Process application" is a highly-sensitive task annotated with the annotation **H.S** and, should be deployed in a Cloud with high level of trust. The "compute installment" task is a sensitive task annotated with **S**, because the bank do not want to expose, on what the installments computation is based. Furthermore, the "retrieve data" task does not reveal any information about the bank strategy, but it produces the data in input to the "Compute installment" task, hence, these two tasks are complementary and have to be separated. Finally, the tasks "Notify eligibility" and "Notify rejection" are not sensitive (they do not encapsulate any of the bank know-how and, do not manipulate any sensitive data) but, if these two tasks are deployed with the highly sensitive task "Process application" they could reveal in which case the bank accepts or rejects a loan application. Hence, these two tasks are complementary with the "process application" task and should be separated.

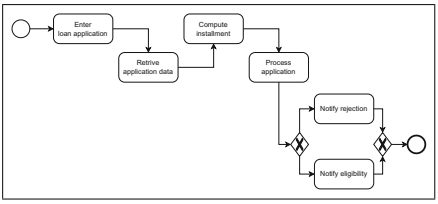

Fig. 7. Bank's loan process

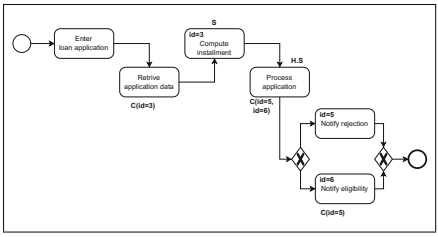

Fig. 8. Annotated loan process

Fragmentation. Following the constraints and rules detailed in Sect. 5.2, we obtain:

– Deploy "Process application" task in a Cloud with a high-level of trust. Let L1 be the level of trust determined by the designer, hence Trust-level(C3) **>=** L1;

– Separate ("Compute installements", "Process application");
– Separate ("Retrieve data", "Compute installments");
– Separate ("Process application", "Notify eligibility");
– Separate ("Process application", "Notify rejection");
– Separate ("Notify rejection", "Notify eligibility");
– Co-locate ("Enter loan application", "Retrieve data").

After that, given these constraints and the co-locate rule, the output of the application of the Algorithm 1 detailed in Sect. 5.3 would be as follows: (**Cloud1**, "Enter loan application", "Retrieve application data", "Notify rejection"), (**Cloud2**, "Compute installments", "Notify eligibility"), (**Cloud3**, "Process application").

The set of tasks for each cloud constitute a fragment of the business process and these latter exchange with the communication tasks Send and Receive (Fig. 9).

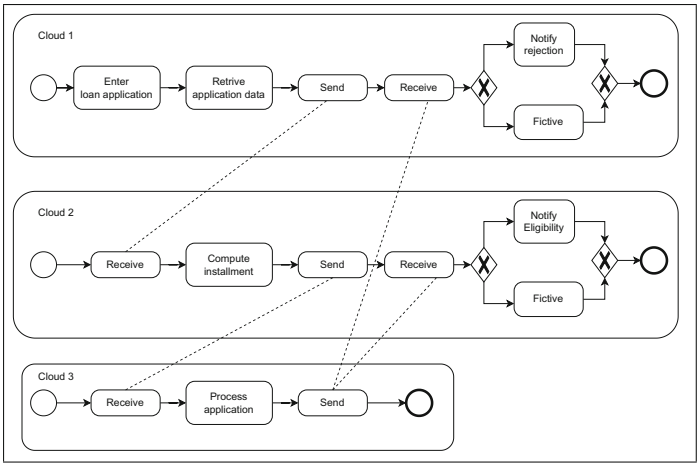

Fig. 9. The loan process fragments according to the hybrid security solution

Some Implementation Details: The implementation of the proposed approach is according to the architecture depicted in Fig. 10. In this architecture, the system is composed of the following components:

– **Security Annotation Tool**: This tool allows to annotate a business process by a designer and generates security rules and constraints.
– **Constraint Solver**: This component takes the generated security rules and constraints as input and checks their consistency. In our case we used the SAT solver MiniSAT[2].

[2] http://minisat.se/MiniSat.html.

– **Task Allocator**: This component takes the set of consistent constraints and allocates the tasks of the business process to the appropriate Clouds.
– **Multi-cloud environment**: This component represents the environment where the business process' tasks will be deployed. We simulated a multi-cloud environment using the container Docker[3]. We also relied on Jenkins[4] in order to automatically deploy the tasks in their respective virtual machines. Lastly, the tasks exchange using the Rest API[5].

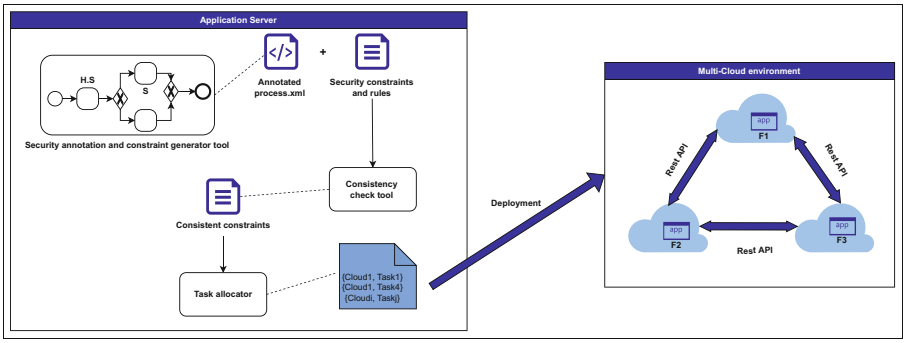

Fig. 10. Deployment architecture of the hybrid security solution

7 Results and Discussion

In this section, we discuss and compare our proposal to the one in [9]. The comparison is proceed according to two criteria *Security* and *Cost*.

7.1 Security

Figure 11, depicts the fragmentation and deployment of the loan business process detailed in Sect. 6, following the solution proposed in [9] (using the obfuscation technique). We can notice that, like with our proposed solution, the deployment requires three distinct Clouds, but in the case of the obfuscation technique the logic of the loan process is half preserved. Indeed, one Cloud has already 80% of the bank know-how through the tasks "Compute installment" and "Process Application" and this is due to the fact that sequential BPMN pattern is not taken into consideration. In this case, one can easily notice that the bank know-how is more secure with the annotation-based solution.

[3] https://www.docker.com/.
[4] https://www.jenkins.io/.
[5] https://www.redhat.com/en/topics/api/what-is-a-rest-api.

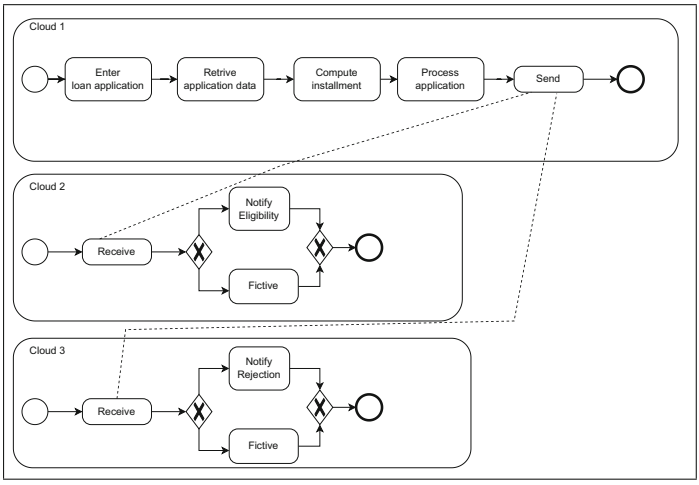

Fig. 11. Loan process fragments according to the obfuscation technique

7.2 Cost Evaluation

Costs (monetary and performance costs) are one of the reasons that makes companies postpone the security requirements of a system to be developed. That is why it is important to find a balance between security and these costs. In this section, we are interested in the monetary costs induced by the proposed solutions. Hence, we propose a cost model to estimate the deployment costs as the sum of the following resources:

- CPU usage: corresponds to the CPU cores needed to execute the process' tasks (CPU/GHz).
- RAM usage: corresponds to the size (in GB) of the needed RAM.
- Data transfer: corresponds to the amount of the exchanged data between the Clouds fragments. (Data transfer/GB)

Both solutions consist in splitting the business process and deploy it in a multi-cloud environment. If we resume the bank account process presented in Sect. 3, securing its know-how with the solutions proposed in [9] requires its deployment into three distinct Clouds. However, as depicted in Figs. 12 and 13, with the hybrid approach the deployment of **the bank account BP**(already described in Fig. 3 in Sect. 6) requires two clouds. Hence, if we consider the proposed Cost model with an example of three Cloud provider's pricing given in Table 2, and supposing that fragment1, fragment2 and fragment3 (generated in case of the obfuscation technique) will be deployed in Cloud1, Cloud2 and Cloud3 respectively, the deployment of the bank account process will cost 423.23$ with the obfuscation technique and 262.45$ with the hybrid approach. Thus, we can notice that the fragmentation according to the obfuscation technique proposed in [9] will generates unnecessary deployment costs whereas as stated in Sect. 3, the solution will not bring a gain regarding the process know-how.

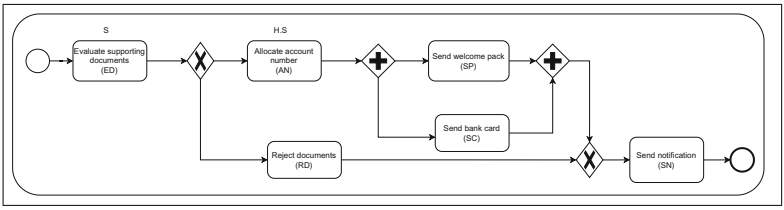

Fig. 12. Annotated Bank Account process

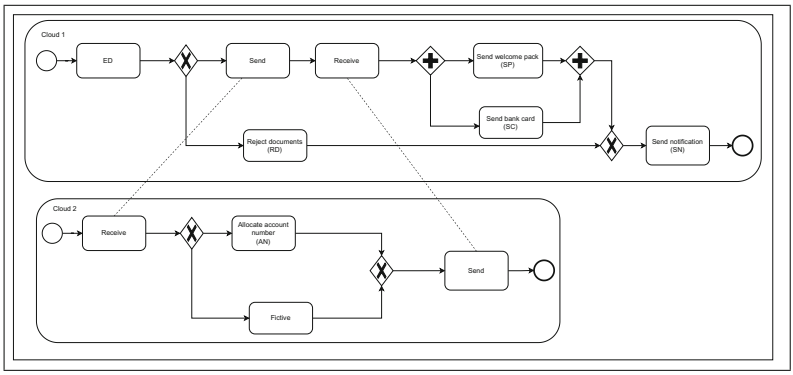

Fig. 13. Bank account process fragments

7.3 The Approach Limitations

Even though this approach ensures the security of the logical aspects of business processes, it has some limitations. In particular, the number of Clouds to be used to deploy the fragments of the business process, relates mainly to the number of highly-sensitive and sensitive tasks which could be quite high depending on the size of the business process in question. Hence, in this work, we are tackling business processes with a moderate number of highly-sensitive and sensitive tasks. However, to deal with larger number of highly-sensitive and sensitive tasks and, in order to reduce the number of the considered clouds, we are currently working on updating the Separate constraints to consider the structure of the business process. Specifically, we argue that the sequential order of the tasks

Table 2. Cloud provider's pricing

Cloud provider	CPU usage ($/GHz/mo)	RAM usage ($/GB/mo)	Data transfer ($/GB)
Cloud1	20.56	4.20	0.02
Cloud2	25.46	4.23	0.04
Cloud3	30.67	5.11	0.08

could play a role in the Separate constraint and the co-locate rule. Hence, we can for example co-locate two sensitive tasks that are not successive. Overall, this work is ongoing and we will provide further details on our findings in future updates.

8 Conclusions and Future Work Directions

This work focuses on enhancing the security of the logical aspect of business processes. Building upon our previous work in [8], where we provided an overview of proposed mechanisms for ensuring business process security, this study specifically examines the obfuscation technique and identifies its limitations in detecting sensitive tasks. We argue that the semi-automatic detection of sensitive tasks proposed in [9] is not always valid. In fact, we demonstrated through business processes examples the fact that sensitive tasks are not always bound to the X(Or) split and the And join gateways. Hence, the human intervene from the start is necessary to specify which task is sensitive and which one is not, according to our definition of sensitive tasks. As an alternative, we proposed an hybrid approach based on textual annotations where we distinguished four types of tasks namely highly-sensitive, sensitive, complementary and non-sensitive, and we adapted the security constraints proposed in the obfuscation technique in order to separate two highly-sensitive tasks, two sensitive tasks, a highly-sensitive task and a sensitive task and a task for its complementary tasks. Even though the obfuscation technique proposed in [9] showed its efficiency, in some cases it can be costly and less effective regarding security. In those cases, the proposed hybrid approach is preferable. In addition of updating the Separate constraints as discussed in Sect. 7.3, we plan to update our algorithm to take into consideration the deployment costs. Indeed, a task Tx could be deployed into different clouds without violating the security deployment constraints. However, regarding monetary costs, one deployment could be preferable to another. Finally, as future work, we intend to explore the impact of business processes changes at runtime on the security constraints and the deployed fragments.

References

1. Ahmed, N., Matulevičius, R.: Securing business processes using security risk-oriented patterns. Comput. Standards Interfaces **36**(4), 723–733 (2014)
2. Altuhhov, O., Matulevičius, R., Ahmed, N.: An extension of business process model and notation for security risk management. Int. J. Inf. Syst. Model. Des. (IJISMD) **4**(4), 93–113 (2013)
3. Argyropoulos, N., Kalloniatis, C., Mouratidis, H., Fish, A.: Incorporating privacy patterns into semi-automatic business process derivation. In: 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS), pp. 1–12. IEEE (2016)
4. Bell, D.E., LaPadula, L.J.: Secure computer systems: mathematical foundations. Technical report, Mitre Corp, Bedford (1973)

5. Dubois, É., Heymans, P., Mayer, N., Matulevičius, R.: A systematic approach to define the domain of information system security risk management. In: Nurcan, S., Salinesi, C., Souveyet, C., Ralyté, J. (eds.) Intentional Perspectives on Information Systems Engineering, pp. 289–306. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12544-7_16
6. Duipmans, E.F., Ferreira Pires, L., Bonino da Silva Santos, L.O.: A transformation-based approach to business process management in the cloud. J. Grid Comput. **12**(2), 191–219 (2014)
7. Enireddy, V., Somasundaram, K., Prabhu, M.R., Babu, D.V., et al.: Data obfuscation technique in cloud security. In: 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), pp. 358–362. IEEE (2021)
8. Farah, A., Saida, B., Mourad, O.C.: On the security of business processes: classification of approaches, comparison, and research directions. In: 2021 International Conference on Networking and Advanced Systems (ICNAS), pp. 1–8. IEEE (2021)
9. Goettelmann, E., Ahmed-Nacer, A., Youcef, S., Godart, C.: Paving the way towards semi-automatic design-time business process model obfuscation. In: 2015 IEEE International Conference on Web Services, pp. 559–566. IEEE (2015)
10. Goettelmann, E., Fdhila, W., Godart, C.: Partitioning and cloud deployment of composite web services under security constraints. In: 2013 IEEE International Conference on Cloud Engineering (IC2E), pp. 193–200. IEEE (2013)
11. Han, Y.B., Sun, J.Y., Wang, G.L., Li, H.F.: A cloud-based BPM architecture with user-end distribution of non-compute-intensive activities and sensitive data. J. Comput. Sci. Technol. **25**(6), 1157–1167 (2010)
12. Jensen, M., Schwenk, J., Bohli, J.M., Gruschka, N., Iacono, L.L.: Security prospects through cloud computing by adopting multiple clouds. In: 2011 IEEE 4th International Conference on Cloud Computing, pp. 565–572. IEEE (2011)
13. de Leoni, M., Dumas, M., García-Bañuelos, L.: Discovering branching conditions from business process execution logs. In: Cortellessa, V., Varró, D. (eds.) FASE 2013. LNCS, vol. 7793, pp. 114–129. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37057-1_9
14. Menzel, M., Warschofsky, R., Meinel, C.: A pattern-driven generation of security policies for service-oriented architectures. In: 2010 IEEE International Conference on Web Services, pp. 243–250. IEEE (2010)
15. Nacer, A.A., Godart, C., Rosinosky, G., Tari, A., Youcef, S.: Business process outsourcing to the cloud: balancing costs with security risks. Comput. Ind. **104**, 59–74 (2019)
16. Nacer, A.A., Godart, C., Youcef, S., Tari, A.: A metric for evaluating the privacy level of a business process logic in a multi-cloud deployment. In: 2017 IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC), pp. 153–158. IEEE (2017)
17. Nacer, A.A., Goettelmann, E., Youcef, S., Tari, A., Godart, C.: Obfuscating a business process by splitting its logic with fake fragments for securing a multi-cloud deployment. In: 2016 IEEE World Congress on Services (SERVICES), pp. 18–25. IEEE (2016)
18. Povoa, L.V., de Souza, W.L., Pires, L.F., do Prado, A.F.: An approach to the decomposition of business processes for execution in the cloud. In: 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), pp. 470–477. IEEE (2014)
19. Rodríguez, A., Fernández-Medina, E., Trujillo, J., Piattini, M.: Secure business process model specification through a UML 2.0 activity diagram profile. Decis. Support Syst. **51**(3), 446–465 (2011)

20. Turki, S.H., Bellaaj, F., Charfi, A., Bouaziz, R.: Modeling security requirements in service based business processes. In: Bider, I., et al. (eds.) BPMDS/EMMSAD 2012. LNBIP, vol. 113, pp. 76–90. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31072-0_6
21. Varela-Vaca, A.J., Warschofsky, R., Gasca, R.M., Pozo, S., Meinel, C.: A security pattern-driven approach toward the automation of risk treatment in business processes. In: Herrero, Á., et al. (eds.) International Joint Conference CISIS'12-ICEUTE'12-SOCO'12 Special Sessions. AISC, vol. 189, pp. 13–23. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-33018-6_2
22. Watson, P.: A multi-level security model for partitioning workflows over federated clouds. J. Cloud Comput. **1**(1), 1–15 (2012). https://doi.org/10.1186/2192-113X-1-15