Anonyfrag: *An Anonymization-Based Approach For Privacy-Preserving BPaaS*

Mehdi Bentounsi*
Université Paris Descartes
Sorbonne Paris Cité, France
mehdi.bentounsi@
parisdescartes.fr

Salima Benbernou†
Université Paris Descartes
Sorbonne Paris Cité, France
salima.benbernou@
parisdescartes.fr

Mikhail J. Atallah‡
Departement of Computer
Science, Purdue University
West Lafayette, IN 47907 USA
mja@cs.purdue.edu

Cheikh S. Deme
SOMONE France
csdeme@somone.fr

ABSTRACT

Many cloud providers offer on demand applications as BPaaS "Business Process as a Service" through multi-tenant cloud platforms, allowing many companies to outsource their business processes. That is, an increasing amount of personal data involved in business processes is automatically gathered and stored on the cloud. For cost saving, some fragments of business processes can be reused i.e., shared between the clients on the cloud regardless of privacy risks. In this paper, we propose an anonymization-based approach to preserve the client business activity while sharing process fragments between organizations on the cloud.

Categories and Subject Descriptors

K.4.1 [**Computers and society**]: Public Policy Issues—*Privacy*

General Terms

Security

*Mehdi Bentounsi is a Research Engineer in SOMONE France. This work is supported by French Association Nationale de la Recherche et de la Technologie under CIFRE contract 1169/2010.

†Portions of this work were supported by Université Paris Descartes collaborative Project Brainshare.

‡Portions of this work were supported by National Science Foundation Grants CNS-0915436, CNS-0913875, Science and Technology Center CCF-0939370; by an NPRP grant from the Qatar National Research Fund; by Grant FA9550-09-1-0223 from the Air Force Office of Scientific Research; and by sponsors of the Center for Education and Research in Information Assurance and Security.

Cloud-I'12, August 31 2012, Istanbul, Turkey

Keywords

Cloud privacy, business process, anonymity

1. INTRODUCTION

Cloud computing is revolutionizing the computer world by allowing the outsourcing of IT infrastructure to specialized providers, similar to the way companies outsource the production of electricity to power utilities. The key driving forces behind cloud computing are the ubiquity of broadband and wireless networking, falling storage costs, and progressive improvements in Internet computing software. The benefits of cloud computing include pay-per-use, reduced power consumption, server consolidation, and more efficient resource utilization. Hence, cloud-service clients will be able to add more capacity at peak demand, reduce costs, experiment with new services, and remove unneeded capacity, whereas service providers will increase utilization via multiplexing, and allow for larger investments. There is great interest in cloud computing from both academic and private research centers. Many providers (Google, Amazon, IBM and Microsoft) now offer cloud solutions at competitive prices.

Cloud services may provide clients with operations at multiple levels of system abstraction and their interfaces come in a consequent variety of shapes and sizes. Some common types of cloud services, and large-scale examples of them, are summarised in Figure 1.

Infrastructure as a Service level (IaaS), like Amazon's EC2, Rackspace, and Nimbus, includes communication network and hardware infrastructure (servers, local, etc.) managed by IT providers. In addition to the hardware platform, in *Platform as a Service* level (PaaS) the operating system is preinstalled with all the necessary middleware for a deployement of business processes such as BPEL Engine and DBMS, etc. PaaS are usually offered as virtual servers (virtualization) on a single physical server. Google's App Engine and Microsoft's Azure are examples of the platform model. *Software as a Service* level (SaaS) offers us a complete and pre-designed service, where the users access with authentication protocols and use services maintained by providers. Salesforce.com has been employing the software level of abstraction.

Furthermore, the cloud model gives the opportunity to

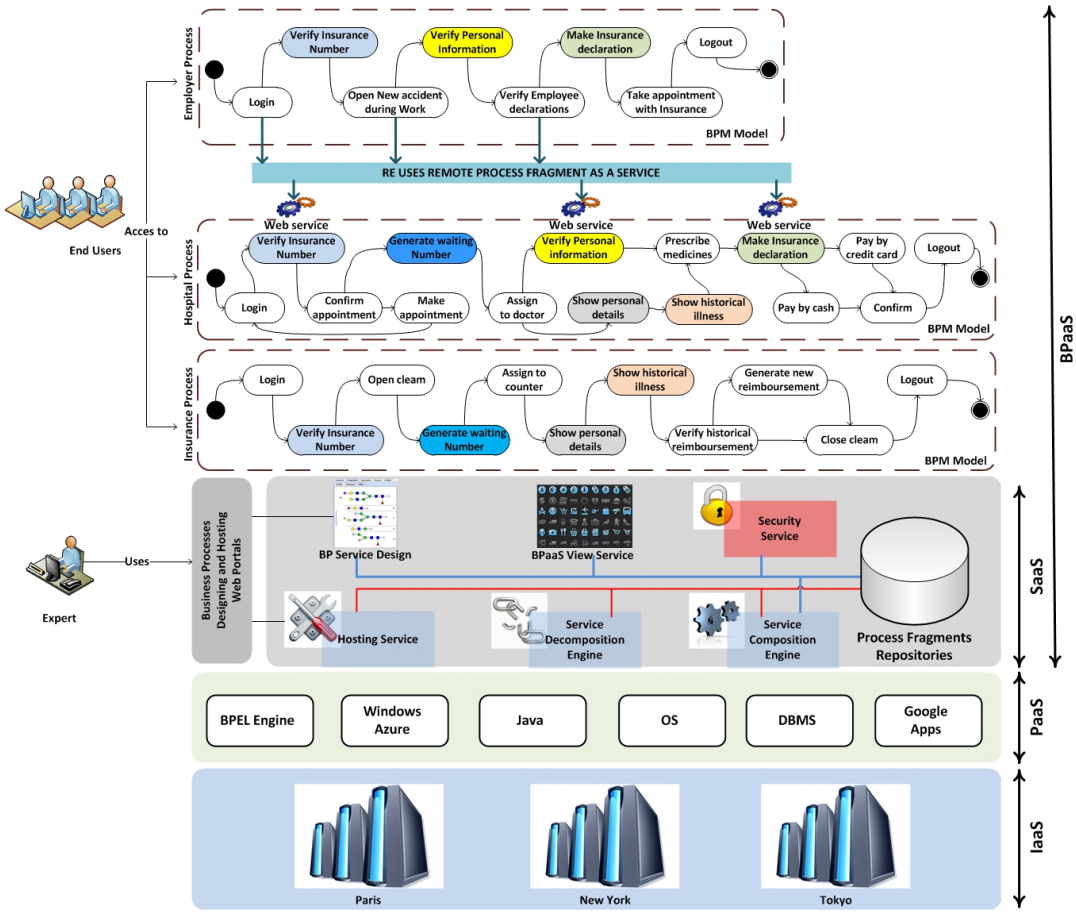

Figure 1: Multi-tenant BPaaS Platform

mash up and compose data and services from a variety of cloud providers to create what's known as a *cloud syndication*. Cloud syndications are essentially federations of cloud providers whose services are aggregated in a single pool [17]. Cloud syndications at the SaaS level are termed "*Business Process as a Service*" (BPaaS). It allows creating unique end-to-end business processes that are usually syndicated with other external services (possibly provided by diverse XaaS providers). BPaaS is emerging as the next major category of cloud IT. By 2015, 50 percent of new *Business Process Outsourcing* (BPO) deals will be delivered as BPaaS (i.e., they will be significantly cloud enabled) [15]. One of the characteristics of cloud model is *multi-tenancy*. Multi-tenancy means that multiple customers (tenants) are served concurrently by one or more hosted application instances [22]. For that, a complex business process deployed by a tenant through the BPaaS can be broken down into smaller (and more manageable) process fragments suitable for re-use as building block in future process modeling. Hence, a process fragment can be seen as a group of connected process elements with high potential for reusability in modeling new business processes by other tenants (avoiding reinventing the wheel).

Moreover, reusing process fragment leads to disclosure of the business activity of the tenants [4]. In fact, there has been a great deal of hype about cloud computing, promising infinite scalability and high availability at low cost, but without addressing security and privacy risk issues. Hence, as of now, companies have only partially overcome their fears about data security: as a consequence of multi-tenant platforms, they have begun to outsource only their *non-strategic* applications, in order to focus on their core business information system (see, e.g., [13]). However, when companies outsource applications, they share IT resources with other companies, creating risks and threats that must therefore be managed if the benefits of the technology are to be achieved.

In order to manage outsourcing in a structured way, with maximum positive outcomes and minimal risks, the *multi-tenant application's server* used in cloud platforms must be able to preserve the *privacy of organizations* when applications and business processes are executed. Indeed, executives now generally recognize that business processes are not just applications that support background workflows, but are also, and more fundamentally, direct generators of revenue and key enablers of strategy. The competitive business environment has forced companies to be operationally innovative in order to outperform their competitors. Innovations in business process design often represent a significant proportion of costs. For, Process Designers may wish to keep some *information confidential*. For example, details of how certain process fragments are composed or the list of process fragments used by an organization may be proprietary. The question is *Why do organizations want to protect this sensitive information ?* Simply in order to preserve their

value-added, and to avoid competitors to steal innovations in business process design. The fact that an adversary may be able to see such confidential data of organizations, is an unacceptable breach of their privacy.

In this paper, we investigate *how to preserve business activity while multi-tenants share a BPaaS in cloud platform*. For that, we propose an anonymization-based approach to hide the provenance of process fragments. We make the following contributions:

BPaaS model. We provide a graph-based approach to model a business process outsourcing as a BPaaS.

Process fragment reusing. Proper reuse of process building artifacts will contribute to cost savings, an increase in the quality of the business processes, and higher productivity in designing the business processes for a tenant.

Privacy-preserving business activity. Reusing process fragments may contribute to disclose business activity. We introduce a new anonymization approach for preserving business activities of the tenants towards hidden process fragments provenance.

The rest of the paper is organized as follows: After describing the problem statement through a motivating example in section 2, in section 3 we provide a BPaaS model for multi-tenant process fragments. Section 4 discusses the anonymization model for privacy-preserving business activities. Section 5 presents an overview of existing works and concludes the paper.

2. PROBLEM STATEMENT

Let us consider three business processes represented by a set of activities using the multi-tenant BPaaS as depicted in Figure 1:

- *Hospital business process* used by hospital staff (e.g. doctors, nurses etc.) to manage consultations and patient history.
- *Insurance business process* used by insurance staff to manage employees claims.
- *Employer business process* used by the human resources department to manage accidents at work.

We assume these business processes are physically deployed in the same BPaaS (e.g. Paris datacenter). Hospital and insurance organizations were the first using the multi-tenant BPaaS through the Business Process Hosting Service to outsource their business processes. As previously explained, a business process is broken down into smaller parts suitable for re-use as building block in future process modeling. Such parts are called *process fragments*. Techniques exist for business process decomposition [14, 2] and fragment identification [11]. In this paper, we assume that the both are done and executed by the Business Process Decomposition Service as depicted in Figure 1.

Over time, the BPaaS provider will build repositories of hundreds or even thousands of process fragments deriving from its customers and running in its datacenter. Duplicate fragments (also called clones) can appear in repositories [21]. Clones are a set of process fragments having the same activity requirement termed *abstract fragment*. For instance, the insurance business process is decomposed in a set of process fragments:

- *"Verify insurance number"* process fragment is deployed by the insurance organization and can be reused in the future to develop employer business process.
- *"Generate waiting number, show personal details* and *show historical illness"* process fragments have clones in the BPaaS deployed by hospital organization.

Design and developement of business processes often represent a significant proportion of costs. For cost saving and in order to avoid building from scratch new business processes, BPaaS may allow customers reusing process fragments as Web Services (i.e., through URLs) [4, 20]. To facilitate reusing process fragments across the BPaaS, each fragments owner should accept that other tenants may reuse some of them. In return, BPaaS must ensure and preserve business activities of organizations.

Taking the example of "Employer" in Figure 1, that wants to design and develop process-based service compositions using BPaaS. Based on the view generated by BPaaS View Service showing a set of process fragments suitable for re-use across the BPaaS. A BPEL expert can reuse: *1) verify insurance number*, *2) verify personal information* and *3) make insurance declaration* process fragments in order to build up the employer business process. The view contains only process fragments from hospital and insurance organizations. By the fact of re-using process fragments, an adversary can easly discover the provenance of the process fragments, and can also deduce the business activity of hospital and insurance organizations. So reusing process fragments can let an adversary learn business activities of BPaaS tenants.

A solution may be to hide "process fragment ID" and "process fragment URL" in BPaaS views. However, merely hiding sensitive information is not sufficient to protect a tenant's business activity. Because an adversary can make link with external informations (e.g. list of organizations using the BPaaS) and discover the provenance of a reused fragment. In order to preserve business activities of organizations, we provide a BPaaS Security Service [4]. It is an anonymization-based approach, inspired by k-anonymity model in databases [19], to hide the provenance of process fragments suitable to be reused (i.e., to make each process fragment indistinguishable from a large enough set of others process fragments). If a tenant were to reuse existing process fragments, he would not know to which company they belong to.

3. PROCESS FRAGMENT

In this section we will introduce some basic notation that will be used in the remainder of the paper. We will also discuss how a BPaaS can let experts design and develop business processes by reusing fragments.

3.1 The formal model

Based on the work in [3], where the authors model a business process as a *directed labeled graph*. We enrich it with the definition of process fragments and BPaaS. We assume the existence of two domains $\mathcal{N}$ of nodes, and $\mathcal{L}$ of node labels. $\mathcal{L}$ is the *disjoint union* of several domains including data values, attribute names, data element names, and activity names. Formally,

DEFINITION 3.1. ***(Business graph)***.
A business graph is a pair $\mathcal{G} = (G, \Gamma)$, where: $G = (N, E)$ is a directed graph in which $N \subset \mathcal{N}$ is a finite set of nodes, E is a set of edges with endpoints in N; and $\Gamma : N \to \mathcal{L}$ is a labeling function for the nodes. Depending on their label type, we refer to the nodes in $\mathcal{G}$ as data element names, data attribute, data value, activity name*, etc.*

DEFINITION 3.2. ***(Business process)***.
A business process (BP for short) is a triple $p = (\mathcal{G}, start, end)$, where: $\mathcal{G}$ is a business graph; start, end are two distinguished activity nodes in $\mathcal{G}$; and each activity node in $\mathcal{G}$ resides on some path from start to end.

A BPaaS is a set of BPs with identifiers. We assume the existence of two domains $\mathcal{P}$ of BPs, and $\mathcal{I}$ of BP's identifiers. Then formally,

DEFINITION 3.3. ***(Business process as a service)***.
A BPaaS model is a pair $\mathcal{S} = (P, \Theta)$, where: $P \subset \mathcal{P}$ is a finite set of BPs deployed on the BPaaS, $P = (p_1, p_2, .., p_i)$; and $\Theta : \mathcal{P} \to \mathcal{I}, \Theta(p) = \mathcal{I}_p$ is an identification function for the whole BPs. Depending on the tenant deploying the BP, we identify a BP in $\mathcal{S}$ by $BP_{tenant_{id}}$.

The BPaaS shown in Figure 2 depicts some identique parts in BPs called *process fragments*. Like a BP defined previously, we model a process fragment as a *directed labeled proper subgraph*, along with a function providing a set of clones (see [21] for a recent paper on the topic).

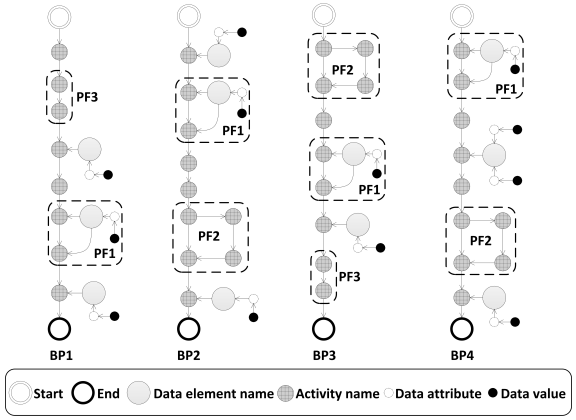

Figure 2: Multi-tenant PF in the BPaaS

We assume the existence of two domains $\mathcal{F}$ of PFs, and $\mathcal{A}$ of abstract fragments (i.e., activity requirements of process fragments). Like instances and classes respectively in object-oriented programming. Two instances of the same class are clones. Then formally,

DEFINITION 3.4. ***(Business subgraph)***.
$\mathcal{H}$ is said a business subgraph of $\mathcal{G}$ (written $\mathcal{H} \subseteq \mathcal{G}$) iff:

- *$N(\mathcal{H}) \subseteq N(\mathcal{G})$, where $N \subset \mathcal{N}$ is a set of nodes; and*
- *$E(\mathcal{H}) \subseteq E(\mathcal{G})$, where E is a set of edges; and*
- *$\Psi_{\mathcal{H}}$ is the restriction of $\Psi_{\mathcal{G}}$.*

When $\mathcal{H} \subseteq \mathcal{G}$ but $\mathcal{H} \neq \mathcal{G}$, we write $\mathcal{H} \subset \mathcal{G}$ and call $\mathcal{H}$ a business proper subgraph of $\mathcal{G}$.

DEFINITION 3.5. ***(Abstract fragment)***.
An abstract fragment (AF for short) is an activity requirement *of a process fragment.*

DEFINITION 3.6. ***(Process fragment)***.
A process fragment (PF for short) is a pair $f = (\alpha, \Delta)$, where: $\alpha \in \mathcal{A}$ is an AF; and $\Delta : \mathcal{A} \to \mathcal{F}, \Delta(\alpha) = F_\alpha$ is a function providing a set $F_\alpha \subset \mathcal{F}$ of business proper subgraphs having the same abstract α.
If the cardinality $|F_\alpha| > 1$, then f is a multi-tenant PF with $|F_\alpha|$ clones : $f^{p_1}, f^{p_2}, .., f^{p_{|F_\alpha|}}$ in $\mathcal{S}$. F is a set of all PFs in $\mathcal{S}$.

To make a difference between BPs and PFs as depicted in Figure 2, each BP in BPaaS is represented by a business graph plus two states *start* and *end*.

EXAMPLE 1. *Let $\mathcal{S}$ be a BPaaS shown in Figure 2, where: $P = \{p_1, p_2, p_3, p_4\}$ is a set of BPs; each BP is identified with an identifier BP_{id}; $F = \{f_1, f_2, f_3\}$ a set of PFs; and $\forall f_i \in F$, we define an AF α_i.*

- *$\Delta(\alpha_1) = \{f_1^{p_1}, f_1^{p_2}, f_1^{p_3}, f_1^{p_4}\}$, we say f_1 is a multi-tenant PF used by all BPs in $\mathcal{S}$.*
- *$\Delta(\alpha_2) = \{f_2^{p_1}, f_2^{p_2}, f_2^{p_3}\}$, we say f_2 is a multi-tenant PF used by three BPs $\{p_1, p_2, p_3\}$ in $\mathcal{S}$.*
- *$\Delta(\alpha_3) = \{f_3^{p_2}, f_3^{p_4}\}$, we say f_3 is a multi-tenant PF used by two BPs $\{p_2, p_4\}$ in $\mathcal{S}$.*

3.2 Reusing Process Fragments

The greatest advantage of using multi-tenant cloud platform is the possibility to share one or a set of PFs. In fact, given a BPaaS $\mathcal{S}$ with some BPs deployed in it, we can develop a new BP by reusing existing PFs as Web Services (How to glue the PFs is out of the scope of the paper).

DEFINITION 3.7. ***(Reusing process fragment)***.
Let us consider: $\mathcal{S} = (\mathcal{P}, \Theta)$ a BPaaS; p a new BP to be developed in $\mathcal{S}$; and $\Omega : \mathcal{F} \to \mathcal{P}$, $\Omega(F) = \acute{p}$ a function performed to develop a new BP $\acute{p}$ by reusing a set F of PFs deployed in $\mathcal{S}$.
In the BPaaS: $\acute{\mathcal{S}} = (\mathcal{P}, \Theta, \Omega)$ where F is a set of PFs, we say that $p \longrightarrow_f \acute{p}$ w.r.t. Ω if $\acute{p}$ is obtained by reusing a PF $f \in F$ in order to develop p.
If $p \longrightarrow_{f_1} \acute{p_1} \longrightarrow_{f_2} \acute{p_2} \longrightarrow_{f_3} ... \longrightarrow_{f_k} \acute{p}$ w.r.t. Ω, then we say that $\acute{p}$ is construction of p by reusing a set $\{f_1, f_2, f_3, ..., f_k\}$ of PFs deployed in $\mathcal{S}$.

The PFR algorithm presents the mechanism for developing new BPs by reusing PFs in BPaaS. Let us consider a new BP depicted in Figure 3 to be build. The construction's process by PFR in detailed in Figure 4. Firstly *(a)*, the BPaaS View Service generates a view of all PFs in the BPaaS suitable for reuse. Secondly *(b)*, the BPEL expert uses BP Service Design and the view generated by BPaaS View Service to develop the new BP.

Note that after one construction step, PF1 fragment has been reused to build BP. At the end of the construction's action steps, two PFs: PF1 and PF2 deployed in BPaaS are reused. Finally *(c)*, the BP Hosting Service deploys BP in the BPaaS and permits connecting to remote PFs as Web Services through their URLs (VM's IP adress + PF's port number). On the one hand, an advantage of BPO through

Algorithm 1 Process Fragment Reusing (PFR)

Require: p a new BP to be developed in BPaaS.
Ensure: $\acute{p}$ a BP developed by reusing PFs in BPaaS.
1: **for all** PFs f in p **do**
2: **if** $f \in F$ **then**
3: $p \longrightarrow_f \acute{p}$ w.r.t. Ω; {If the PF f is available in BPaaS, it will be reused to develop p}
4: $p \leftarrow \acute{p}$; {Preparation for the next step}
5: **end if**
6: **end for**
7: **return** $\acute{p}$.

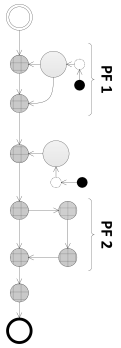

Figure 3: New Business process design

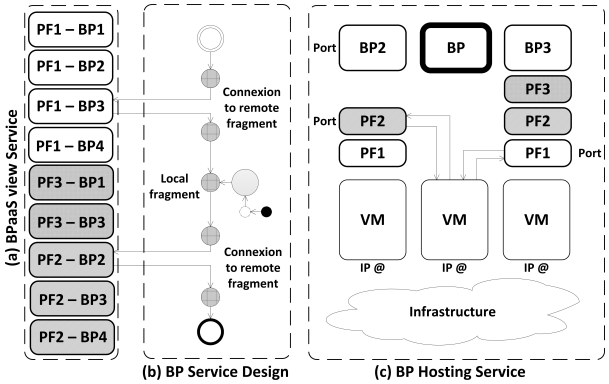

Figure 4: BP's construction in BPaaS

BPaaS is the possibility to develop new BPs from scratch at lower cost. On the other hand, the fact of knowing business activities of other tenants can be seen as an *intrusion* on their privacy and would like to keep it confidential. Hence, reusing PFs in a multi-tenant BPaaS can enable adversaries to learn the business activities of tenants. The deal would allow BPaaS customers reusing PFs to develop new BPs from scratch, and privacy-preserving business activities of PFs owners.

4. PRIVACY-PRESERVING BUSINESS PROCESS OUTSOURCING

4.1 Views on BPaaS

We introduce the notion of a *BPaaS view* (playing a similar role in database [8]) that provides a set of business proper subgraphs having the same AF or clones (e.g., "*Verify insurance number*" fragment in Figure 1). *BPaaS views* will be used as a basis to preserve business activities of tenants from intrusion where their PFs are reused. Formally,

DEFINITION 4.1. ***(BPaaS views).***
Let us consider: $\mathcal{S}$ a BPaaS including a set of BPs $P = \{p_1, p_2, \ldots, p_i\}$; α an AF; and V_α a set of business proper subgraph having the same abstract α.
V_α is called a view *on $\mathcal{S}$ w.r.t. α.*

In the following and in order to manage the views, we define three operations:

DEFINITION 4.2. ***(Operations on BPaaS views).***
Let us consider: $\mathcal{S}$ a BPaaS including a set of BPs $P = \{p_1, p_2, \ldots, p_i\}$; α and β two AFs; V_α (resp. V_β) a view on $\mathcal{S}$ w.r.t. α (resp. β), then :

1. *$V_{\neg\alpha}$ is said a view of $\mathcal{S}$ w.r.t. $\neg\alpha$, iff $V_{\neg\alpha}$ contains all business proper subgraphs in $\mathcal{S}$ **not** having the AF α (Negation).*
2. *$V_{\alpha\wedge\beta}$ is said a view of $\mathcal{S}$ w.r.t. $\alpha \wedge \beta$ iff $V_{\alpha\wedge\beta}$ contains all business proper subgraphs in $\mathcal{S}$ having the AFs α **and** β (Conjonction).*
3. *$V_{\alpha\vee\beta}$ is said a view of $\mathcal{S}$ w.r.t. $\alpha\vee\beta$, iff $V_{\alpha\vee\beta}$ contains all business proper subgraphs in $\mathcal{S}$ having the AFs α **or** β (Disjonction).*

In the following propositions, we note: $\mathcal{S}$ is a BPaaS, V_α (resp. V_β) is a view on $\mathcal{S}$ w.r.t. α (resp. β) .

PROPOSITION 1. *$V_{\neg\alpha} = \mathcal{S} \setminus V_\alpha$, $V_{\neg\alpha}$ is an* absolute complement of a view *V_α in $\mathcal{S}$.*

PROPOSITION 2. *$V_{\alpha\vee\beta} = V_\alpha \cup V_\beta$, is an* union of views *in $\mathcal{S}$.*

4.2 Anonymous views on BPaaS

Given a BPaaS $\mathcal{S}$ with a set of PFs F. The BPaaS would like to protect against linking a PF to a tenant in $\mathcal{S}$. As mentioned previously, PFs can be reused when building new BPs. The fact to know the owners of some PFs may disclose the business activity of tenants. A set of PFs enabling an adversary to learn a tenant's business activity is termed Quasi-identifier fragments. Formally,

DEFINITION 4.3. ***(Quasi-identifier fragments).***
A set $A = \{\alpha_1, \alpha_2, \ldots, \alpha_i\}$ of AFs in BPaaS $\mathcal{S}$ is called Quasi-identifier fragments, if these AFs can be used to learn the tenants business activities in the BPaaS.

One example of a quasi-identifier fragment is "*verify insurance number*" from Figure 1. It can be used to learn the business activity of hospital organization in the BPaaS. An adversary can discover that the hospital organization outsources a BP to BPaaS, and can also make a link between a patient and the hospital if he recovers an input/output value (e.g. the social security number of a patient). Let us denote the set of all quasi-identifier fragments by $\mathcal{QIF}$.

Inspired by anonymity technique for privacy-preserving in databases [19], we define $K_l - anonyfrag$: an anonymity model for process fragments, consists in generating anonymous views on the BPaaS.

During the definition of a PF, we said the PFs could be multi-tenants. We consider a BPaaS $\mathcal{S}$ with l tenants. The $k_l - anonyfrag$ requirement below, which states that in every view on BPaaS we have at most K clones. Otherwise, there exists at most K different business proper subgraphs in $\mathcal{S}$ implementing the same AF in $\mathcal{QIF}$.

DEFINITION 4.4. ***($\mathbf{K_l}$ − **anonyfrag** requirement).***
$K_l - anonyfrag$ requirement is each view V_α on BPaaS w.r.t. $\alpha \in \mathcal{QIF}$ must contain at most K clones.

Since it seems impossible or highly impractical and limiting to make assumptions on PFs to adversaries to discover business activities of tenants when reusing a PF to construct a new BP. In the following, we define a $K_l - anonyfrag$:

DEFINITION 4.5. ***($\mathbf{K_l}$ − **anonyfrag**).***
Given a BPaaS $\mathcal{S}$ used by l tenants; and an AF α with a set of at most K business proper subgraphs or clones in $\mathcal{S}$. An adversary knows that it exists at most K clones implementing α are deployed in $\mathcal{S}$; and doesn't know :

1. *Exactly the number of tenants that own the K business proper subgraphs among l tenants.*
2. *Which tenants exactly have deployed the PF in $\mathcal{S}$.*

A view V_F satisfies $K_l - anonyfrag$ if for every AF $\alpha_i \in (\mathcal{QIF} \cap F)$ the cardinality $|V_{\alpha_i}| \in [1, K]$.

Note that if $|V_{f_i}| = K = 1$, it means that only one tenant in the BPaaS among l has deployed the PF f_i. In this case the probability $P_{pro}(f_i)$ that an adversary may discover the provenance of a f_i is minimum:

$$P_{pro}(f_i) = \frac{1}{l}$$

The Anonymized View $V_F^\star$

Since the $\mathcal{QIF}$ might uniquely learn business activities in V_F, the view V_F is not used to construct new BPs; it is subjected to an anonymization procedure and the resulting view $V_F^\star$ is used instead.

The algorithm P3FR depicts the mechanism for privacy-preserving building BPs by reusing PFs in a BPaaS.

Algorithm 2 Privacy-preserving Process Fragment Reusing (P3FR)

Require: p a new BP to be developed in BPaaS.
Ensure: $\acute{p}$ a BP developed by reusing PFs in BPaaS.
1: Compute $V_F^\star$ which satisfies $K_l - anonyfrag$;
2: **for all** PFs f in p **do**
3: **if** f appears in the view $V_F^\star$ **then**
4: Select a business proper subgraph f^{p_i} from F_f;
5: $p \longrightarrow_{f^{p_i}} \acute{p}$ w.r.t. Ω;
6: $p \leftarrow \acute{p}$; {Preparation for the next step}
7: **end if**
8: **end for**
9: **return** $\acute{p}$.

4.3 Running Example

Consider a BPaaS $\mathcal{S}$ shown in Figure 2, with ($l = 4$) tenants ; $P = \{p_1, p_2, p_3, p_4\}$ a set of BPs deployed in $\mathcal{S}$; and $F = \{f_1, f_2, f_3\}$ a set of PFs. We assume that the Quasi-identifier Fragments $\mathcal{QIF} = F = \{f_1, f_2, f_3\}$.

1. Figure 5 shows a $\mathbf{1_4}$ − **anonyfrag** view on BPaaS w.r.t. F. Using DEFINITION 4.5 (of $\mathbf{k_l}$ − **anonyfrag**), we put: $\forall f_i, |V_{f_i}| \in [1, 1]$. In $\mathcal{S}$, all PFs have at least two clones. For this, the view is empty. Note the probability that an adversary may discover the provenance of a PF $P_{pro}(f_i) = \frac{1}{4} = 0.25$ is minimum.

2. Figure 6 shows a $\mathbf{2_4}$ − **anonyfrag** view on BPaaS w.r.t. F. Note a $\mathbf{2_4}$ − **anonyfrag** view must contain only PFs having at most two clones ($\forall f_i, |V_{f_i}| \in [1, 2]$). In $\mathcal{S}$, only one PF PF3 has two clones and appears in the view. $P_{pro}(f_i) = \frac{2}{4} = 0.5$ is the probability that an adversary may discover the provenance of a PF.

3. In Figure 7, the view on BPaaS w.r.t. F respects the $\mathbf{3_4}$ − **anonyfrag** requirement. This view contain only PFs having at most three clones ($\forall f_i, |V_{f_i}| \in [1, 3]$). Only one PF PF1 has four clones in $\mathcal{S}$ and not appears in the view. $\mathbf{3_4}$ − **anonyfrag** requirement guarantees us a probability $P_{pro}(f_i) = \frac{3}{4} = 0.75$ that an adversary may discover the provenance of a PF.

4. Finally, Figure 8 shows a $\mathbf{4_4}$ − **anonyfrag** view on BPaaS w.r.t. F. All PFs appear in the view. In the example, all PFs have at most four clones. Note $K = l = 4$ and the probability that an adversary may discover the provenance of a PF $P_{pro}(f_i) = \frac{4}{4} = 1$ is maximum. Otherwise, the view does not preserve the business activity of tenants. For this we put the requirement: $\forall f_i, |V_{f_i}| \in [1, K]$ where $K \neq l$.

4.4 Attacks against anonyfrag

Even when sufficient care is take to identify the quasi-identifier fragment, a solution that adheres to $K_l - anonyfrag$ can still be vulnerable to attacks. Two specific attack scenarios are described below. Due to the space limitation, we will not go further far formal aspects. Fortunately, these attacks can be thwarted by due diligence to some accompanying practices, which are also described.

4.4.1 Unsorted matching attack against anonyfrag

This attack is based on the manner in which clones are selected in the view. We omitted the selection's process of a business proper subgraph in views in this discussion, in real-world use this is often a problem (e.g. The selection is based on the most recent fragment). It can be corrected of course, by randomly selecting a business proper subgraph. Otherwise, the construction's action can leak business activities of tenants.

4.4.2 Temporal attack against anonyfrag

Views on BPaaS are dynamic. BPs are added, changed, and removed constantly. As a result of that, releases views can be subject to a temporal inference attack. Let $\mathcal{S}_0$ be the original BPaaS at time $t = 0$. Assume a $k_l - anonyfrag$ view based on $\mathcal{S}_0$, which we will call V_0, is computed. At time t, assume additional BPs were added to the BPaaS $\mathcal{S}_0$, so it comes $\mathcal{S}_t$. Let V_t be a $k_l - anonyfrag$ view based on $\mathcal{S}_t$ that is released at time t. Because there is no requirement that V_0 respect V_t, using the views V_0 and V_t may reveal business activities and thereby compromise $k_l - anonyfrag$ protection.

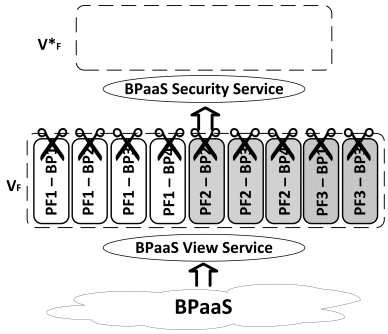

Figure 5: 1_5 – anonyfrag view on BPaaS

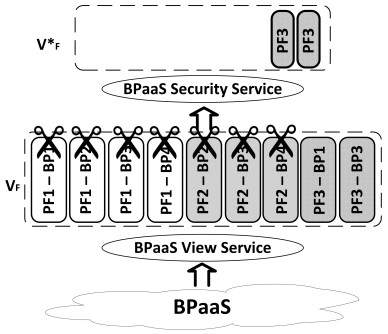

Figure 6: 2_5 – anonyfrag view on BPaaS

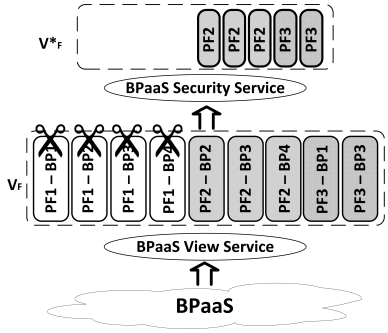

Figure 7: 3_5 – anonyfrag view on BPaaS

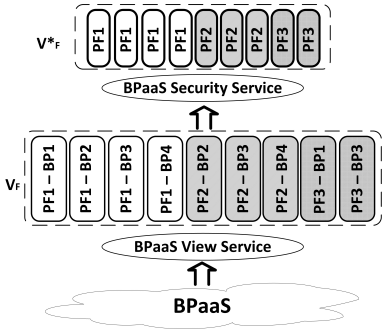

Figure 8: 4_5 – anonyfrag view on BPaaS

4.5 Implementation

We are implementing a prototype of anonyfrag-framework to simulate and show the impact of our approach. Figure 9 depicts a snapshot of Web 2.0 interface that allows BPEL experts to query BPaaS (large process fragment model repositories) through keywords. Anonymous views permit to select process fragments depending on their descriptions and QoS (availability and response time).

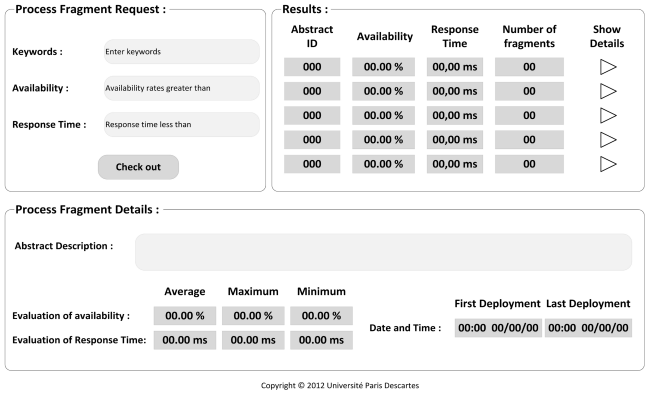

Figure 9: A snapshot of anonyfrag-framework

Randomly generated scenarios are used for the experiments. Each scenario simulates the use of BPaaS by organizations for business process outsourcing. QoS values of availability rates and response time for each process fragment are generated randomly with uniform probability.

5. RELATED WORK

In the literature, some works have been provided for business process modeling such as BP-QL [3] and Business process decomposition [14, 2]. In [14] a BPEL decomposition process results into several partitions assigned to different process engines. The partitions are defined at the design stage of a process where the activities are assigned to the execution sites manually. The aim of such approach is to enhance the execution of the original process. Similarly, an autonomic approach is proposed in [2], to partition a centralized BPEL process into a set of coordinated processes. [21] proposed an indexing structure to support the fast detection of clones (duplicate fragments) in large process model repositories, in order to create or extend new process models.

Nowadays, with emerging technologies like cloud computing, the entreprises / organizations increase drastically their interest for business process outsourcing to BPaaS providers. [1] investigated the execution of BPEL processes in different cloud computing delivery models, and showed security and trust issues affect the business processes outsourcing. Privacy of users in the business processes outsourcing was not yet addressed. In fact, research on data privacy for cloud computing is still in its early stages. The current works are mainly describing the need for security and privacy, and their efforts in defining guidelines [12, 6]. In [12], the authors look at the main security and privacy issues pertinent to cloud computing, as they relate to outsourcing portions of the organizational computing environment. They point out areas of concern with public clouds that require special attention and provide necessary background to make informed security decision. Cloud Data Security project [6] aims to design cloud services adhering to government privacy laws. CloudDataSec introduces a six-layer security model for

cloud computing and three levels of security assurance for enterprises to take advantage of. An encryption-based solution was proposed in [18], the authors describe a Privacy Manager for cloud computing. It is an obfuscation-based approach. The user's private data is sent to the cloud in an encrypted form, and the processing is done on the encrypted data. The output of the processing is de-obfuscated by the privacy manager to reveal the correct result. In [9], the authors state that security and privacy issues of traditional web applications are also valid for cloud environments. Furthermore, they agree that encryption is a feasible solution to security problems in cloud computing. Assuming databases are used to store data in the cloud, they showed that encryption will cause a performance overhead depending on the transaction. Aforementioned works didn't handle the privacy in business processes.

Anonymization-based approach for handling privacy has been proposed in [19], and has been studied in various contexts: data mining [7], social networks [5], statistical databases, etc. However there is very little investigation on privacy in cloud computing area. [10] design a privacy preserved context service protocol in order to protect user privacy from exposed context information. PPCS uses k-anonymity and l-diversity to generate blur context information to make difficult for an attacker to determine information's sources. [16] study and resolve a real-life privacy problem in a data mashup application for the financial industry in Sweden, and propose a privacy-preserving data mashup based k-anonymization algorithm to securely integrate private data from different data providers.

The focus of almost all these works has been set on ensuring security and privacy in cloud computing and none of them adresses in particular business process outsourcing with preserving privacy to get reusable fragments. To the best of our knowledge, our work is the first to address the privacy-preserving Business Process as a Service .

6. CONCLUSION

Cloud computing and Business Process as a Service are new emerging delivery models offering the possibility to Business Process Outsourcing and enabling the enterprises to focus on their competencies. In this paper we investigated modeling multi-tenant process fragments in a BPaaS. At first glance, we provided a formal model for BPaaS. In addition, in order to avoid building a new business process from scratch in the BPaaS, we emphasize the process fragment reusing by the BPaaS provider. Furthermore, when reusing existing business process fragments to design a new business process, an anonymization-based approach is proposed to preserve the privacy of a tenant in the BPaaS regarding its business activity.

7. REFERENCES

[1] T. Anstett, F. Leymann, R. Mietzner, and S. Strauch. Towards bpel in the cloud: Exploiting different delivery models for the execution of business processes. *IEEE Congress on Services*, pages 670–677, 2009.

[2] L. Baresi, A. Maurino, and S. Modafferi. Towards distributed bpel orchestrations. *ECEASST*, 3, 2006.

[3] C. Beeri, A. Eyal, S. Kamenkovich, and T. Milo. Querying business processes. *VLDB*, pages 343–354, 2006.

[4] M. Bentounsi, S. Benbernou, and M. J. Atallah. Privacy-preserving business process outsourcing. *ICWS 2012*, 2012.

[5] A. Campan and T. Truta. Data and structural k-anonymity in social networks. *PinKDD*, pages 33–54, 2009.

[6] F. Dölitzscher, C. Reich, and A. Sulistio. Designing cloud services adhering to government privacy laws. *CIT*, pages 930–935, 2010.

[7] A. Friedman, R. Wolff, and A. Schuster. Providing k-anonymity in data mining. *VLDB J*, 17(4):789–804, 2008.

[8] A. Halevy. Answering queries using views: A survey. *VLDB J*, 10(4):270–294, 2001.

[9] J. Hu and A. Klein. A benchmark of transparent data encryption for migration of web applications in the cloud. *DASC*, 2009.

[10] X. Huang, Y. He, H. Yifan, L. Li, L. Sun, S. Zhang, Y. Jiang, and T. Zhang. Privacy of value-added context-aware service cloud. *CloudCom*, pages 547–552, 2009.

[11] D. Ivanovic, M. Carro, and M. Hermenegildo. Automatic fragment identification in workflows based on sharing analysis. *ICSOC*, pages 350–364, 2010.

[12] A. Jansen, W. Cloud hooks: Security and privacy issues in cloud computing. *HICSS-44*, pages 1–10, 2011.

[13] M. Jensen, J. Schwenk, N. Gruschka, and L. L. Iacono. On technical security issues in cloud computing. *IEEE CLOUD*, pages 109–116, 2009.

[14] R. Khalaf and F. Leymann. E role-based decomposition of business processes using bpel. *ICWS*, pages 770–780, 2006.

[15] R. McNeill. The evolution of business process as a service (bpaas). *Saugatuck Technology Inc.*, October 2010.

[16] N. Mohammed, B. C. M. Fung, K. Wang, and P. C. K. Hung. Privacy-preserving data mashup. *EDBT*, pages 228–239, 2009.

[17] M. P. Papazoglou. Cloud blueprints for integrating and managing cloud federations. *Software Service and Application Engineering*, pages 102–119, 2012.

[18] S. Pearson, Y. Shen, and M. Mowbray. A privacy manager for cloud computing. *CloudCom*, pages 90–106, 2009.

[19] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng.*, 13(6):1010–1027, 2001.

[20] D. Schumm, D. Dentsas, M. Hahn, D. Karastoyanova, F. Leymann, and M. Sonntag. Web service composition reuse through shared process fragment libraries. *ICWE*, 2012.

[21] R. Uba, M. Dumas, L. García-Bañuelos, and M. La Rosa. Clone detection in repositories of business process models. *BPM*, pages 248–264, 2011.

[22] H. Yaish, M. Goyal, and G. Feuerlicht. A novel multi-tenant architecture design for software as a service applications. *CLOSER*, pages 82–88, 2012.