Proceedings of the Federated Conference on Computer Science and Information Systems pp. 925–934
DOI: 10.15439/2017F330
ISSN 2300-5963 ACSIS, Vol. 11

Privacy Preserving BPMS for Collaborative BPaaS

Michael Glöckner*, Björn Schwarzbach*, Sergei Makarov*, Bogdan Franczyk*† and André Ludwig‡
*Leipzig University, Germany
†Uniwersytet Ekonomiczny we Wrocławiu, Poland
‡Kühne Logistics University, Germany
{gloeckner,schwarzbach,makarov,franczyk}@wifa.uni-leipzig.de, andre.ludwig@the-klu.org

***Abstract*—Collaboration in business environments is an ongoing trend that is enabled by and based on cloud computing. It supports flexible and ad-hoc reconfiguration and integration of different services, which are provided and used via the internet, and implemented within business processes. This is an important competitive advantage for the participating stakeholders. However, trust, policy compliance, and data privacy are emerging issues that result from the distributed data handling in cloud-based business processes. Up to now, several architectures and technical systems that enable the cloud-based collaboration within business processes have been developed, but the selection of an appropriate business process management system (BPMS) is missing. An implemented BPMS has to meet certain requirements that result from the cloud-based characteristics and from the other implemented systems. This paper derives requirements for BPMSs in cloud-based environments, currently available BPMSs are evaluated against the derived requirements and the selected one is implemented subsequently.**

I. Introduction

COLLABORATION is one of the essential mechanisms in business environments that follow the trend of outsourcing and concentration on core competencies in order to create customized business processes. Foundation and enabler of this development is the technology of cloud computing (CC) [1], [2]. Next to benefits such as facilitated collaboration, flexibility, and ad-hoc reconfiguration, there are certain challenges arising from the virtualization of resources and the decentralized data handling within the CC paradigm. Especially, the cloud based planning, operation and monitoring of business processes incorporates sensitive data. Hence, data privacy and the related policy compliance results in a crucial challenge for affected companies that participate in such a cloud-based business environment [3]. Recent publications deal with the issues of data privacy in cloud-based business processes [4], based on the so called collaborative business process as a service (BPaaS) [4], [5]. Several components of a comprehensive architecture and their interfaces have been described in several papers [4], [6], [7]. The only missing core component of the presented architecture is the Business Process Management System (BPMS), which the paper focuses on.

The BPMS has to meet several requirements from the perspective of business and privacy policies. Further, the already implemented components of the architecture impose additional requirements based on the given ways of communication and the underlying logic. In summary, the paper answers the following research question: *How is a privacy preserving BPMS to be designed and implemented in order to grant data privacy for collaborative BPaaS?*

The paper is structured as follows: After the introductory motivation, section 2 focuses on the theoretical background. Section 3 briefly introduces the applied methodology, which comprises the deriving of the requirements for privacy-driven cloud-based BPMS, the introduction of several alternatives and their evaluation. After section 4 that briefly introduces the implementation of the selected BPMS, the paper is concluded in section 5.

II. Theoretical Background

Business Engineering (BE) is the approach of the methodical implementation of new business solutions as well as company's organization design, especially of business processes comprising its modeling, strategy and information systems usage [8]. Alphar et. al. [9] refer to St. Gallen's business architecture and define BE as the method of integrated design based on process orientation focusing on the link between business strategy and IT. The key aspects of BE are the optimization of existing and the design of new customer-oriented processes. The design should ensure substantially completed processes and that activities are presented ordered chronologically and logically with regards to the business objectives [9], [10]. The outputs of the activities have an essential impact on the process execution as they serve as input for the particular subsequent activity. The most important challenges and main objectives of business engineering are execution time, promotion of customer loyalty as well as reduction of process faults and errors [8].

There are two fundamental approaches used for process design: top-down and bottom-up. The first one proceeds from business strategy down to its IT-implementation and comprises vision, initial diagnose, redesign, development

The work presented in this paper was funded by the German Federal Ministry of Education and Research within the project 'Logistik Service Engineering und Management' (LSEM) under the reference BMBF 03IPT504X and within the project 'Privacy-preserving Methods and Tools for cloud-based Business Processes' (PREsTiGE) under the reference BMBF 16KIS0082K. More information can be found on the websites http://lsem.de and http://prestige.wifa.uni-leipzig.de.

as well as evaluation phases. The benefits of this approach are the systematic consideration of customer orientation, business strategy, and its objectives. Moreover, it allows to build end-to-end processes starting at customer needs and ending with its achievements [11]. However, [9] argues, that the criticized bottom-up project implementations are frequently applied due to rare possibility of top-down's greenfield approach integration.

The need of proper process design and its preparation results from the information models' complexity and their high quality requirements [12]. A methodical approach supports the handling of complex business model design, as those models consist of multiple, standardized, and modular components and dependencies. Hence, systems and modules can be reused in other business models to accelerate the design process [9]. With the help of reference models, during BE particular business IT architectures can be developed [13]. There is a variety of solutions, notations, and approaches for process modeling such as ARIS (Architecture of Integrated Information Systems), BPMN (Business Process Model and Notation), EPC (Event-driven Process Chain) etc. [8].

Business strategy and business objectives are the foundation and purpose of business process modelling [11]. Business process modeling is part of business process management, which encompasses the whole livecycle of business processes. This comprises design, modeling, execution, monitoring, and optimization [14]. Business process management systems (BPMS) comprise tools, methods, and concepts to support business process management for the whole business process life-cycle [11], [12].

The result of business process design is a machine-readable process description, which can be executed by process engines of the BPMSs. Languages for this purpose are e.g. XPDL (XML (Extensible Markup Language) Process Definition Language) and BPEL (Business Process Execution Language). The major drawback of those languages is the lack in graphical process representation [10], [15], [16]. In order to overcome this problem, the language BPMN was developed. Nowadays, it is the most common notation in business process management due to its multiple advantages: the processes created in BPMN are executable, machine readable, and further a special graphical representation was created to enable human readability. Consequently, the BPMN has prevailed against BPEL and other languages and has become a standard in process modeling. BPMN is based on structured layers, each of those consists of different elements. The most fundamental ones, that make it possible to construct BPMN diagrams are Process, Choreography and Collaboration [17].

Due to the rapid growth of information system's modularity by the implementation of technologies such as SOA and Web Services, BPM is an integrator for coupling and connection of independent components with the help of the process level. This helps to deliver personalized and specialized processes to customers, in order to provide a mix of responsive products and services, that work together in value chains. The cloud provides a flexible and scalable environment for such integration tasks. Hence, the realization of BPMS in the cloud is a consequential step. Benefits of the cloud paradigm's implementation are e.g. reliable performance on demand, scalability, and predictive analytics for process tasks. Usually cloud services are distinguished into Infrastructure as a Service, Platform as a Service, and Software as a Service, which are different levels of abstraction of services from virtualized hardware to whole ready to use software products [18], [19]. Reference [18] defines the three following diverse system categories of BPMS, that could be implemented on the particular cloud level: Interconnecting Systems for information exchange through all accessible channels on IaaS and PaaS; Adaptive Systems for internal process and customer monitoring based on SaaS; Specialized customer-oriented Systems at SaaS level. When it comes to BPMS in the cloud, another layer, the Business Process as a Service (BPaaS) is added to the stack. The mentioned benefits of the cloud paradigm are also valid on BPaaS layer.

Considering how functionality of BPM architecture spreads across the cloud, [20] defines the four following levels: Infrastructure Service Layer, which comprises Service Bus with File Storage System; Platform Service Layer for business process engine and pre-built business process rules libraries with diverse middle-ware core elements; BPMS-as-a-Platform layer, which provide full BPM-life-cycle management support, especially process design and monitoring; and finally, Software and Service Layer with the tools for information analysis on the highest abstract cloud computing level. According to [5], the transformation of BPMS paradigm from domain-specific business process to executable work-flow in Cloud consists of five levels to be fulfilled as follows. Knowledge Externalization comprises the representation of cloud service features in a human and machine readable way, can be achieved by clear semantics and language. BPaas design maps the activities of business process to cloud services. The third level of transformation is BPaaS allocation, where each business process in the cloud can be understood as a service, which can be deployed to the cloud. The BPaaS execution level provides the orchestration of such cloud service on a higher abstraction. Finally, the level of BPaaS evaluation collects all information from the BPaaS deployment in multi-cloud environments to provide conceptual analytics on business level. Through all those levels security and privacy has to be guaranteed.

The public access as well as multi-tenancy provided by cloud computing have a tremendous impact on information security and privacy. Reference [21] defines the five following perspectives based on high level security requirements of business processes, in order to guarantee integrity, consistency and completeness:

- *Information Perspective* refers to the structure and relationship of information units

- *Function Perspective* is the mapping of and the dataflow between process activities
- *Dynamical Perspective* concers the representation of all states for each information unit as well as its status transformation during information unit lifecycle
- *Organization Perspective* provides information about who executes which activity at which point of time
- *Business Process Perspective* represents the business process as activity and information stream from the perspective of the whole process.

Apart from the process security, also security on the cloud environment itself should be considered. Reference [22] argues the communication of components based on Trusted Third Party (TTP) Services as the ideal solution for integrity, confidentiality, and credibility in the cloud. TTPs are operationally connected certificate paths, that provide a notion about Public Key Infrastructure (PKI) and support strong process authentication, authorization for an access to resources, data banks and informative systems as well as data confidentiality.

Summarizing, BPaaS provides a suitable approach for business process management in cloud-based collaborations. One of the major challenges in this field is the compliance of privacy policies due to decentralized handling of sensitive data. This challenge has already been met for some parts of BPaaS architectures [4]. However, the evaluation and selection of a suitable BPMS is still missing. The following section derives requirements for privacy preservation in collaborative BPaaS and evaluates existing BPMS.

III. Evaluation and Selection of BPMS

In this section, a method for BPMS evaluation and selection [23] is introduced and applied in the field of collaborative BPaaS. Requirements of a BPMS in collaborative cloud environments are derived from business perspective and from the existing architecture's perspective. Afterward, existing BPM tools are identified and evaluated with the help of the derived requirements and eventually an appropriate tool is selected for subsequent implementation.

The method of [23] comprises the three steps that are depicted in Figure 1: identifying the the organizational requirements, identifying BPM tools, and selection of appropriate BPM tool with the help of the requirements. The common and specific requirements help to improve the decision of tool selection. According to [23] both first steps are executed in parallel. In the identification of BPM tools during second step, there are some weak spots in the method because of its proposal to identify *all* available market tools without any restriction. This leads to comprehensive and extensive list, that should be elaborated completely. One of the options here is to reduce the list by identifying common criteria both for software and BPM tools with regards to common evaluation practices. During the last step, each BPM tool from the list is evaluated by

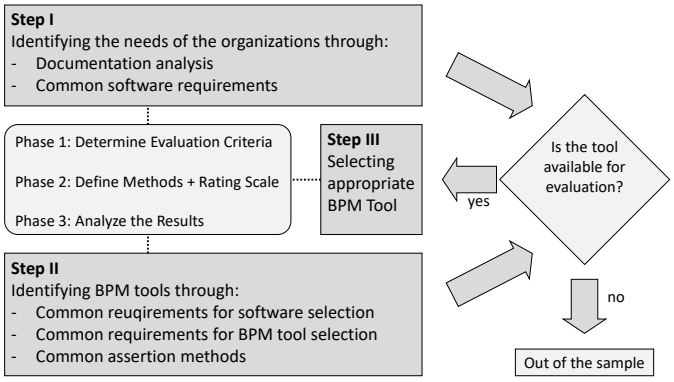

Fig. 1. Methodology for BPM Tool selection (referring to [23], [24])

particular criteria in order to evaluate the tools' suitability to meet the requirements. Moreover, a tool cannot be evaluated, if at least one criteria cannot be applied and evaluated.

A. Identifying Organizational Requirements

According to [24] there are two types of organizational requirements, that are described in functional specifications. First type comprises a compilation with the scope on delivery and performance. The second type describes the implementation. Further, a distinction can be made between functional and non-functional requirements [24]. According to [25] there are various document types for documentation such as use case and class diagrams, software glossary, sequence diagrams, domain models, performance and interface description, etc.

An interview among companies from the logistics sector revealed that privacy is one of the most important concerns of companies when it comes to collaborative BPaaS. Also the application of SOA (Service Oriented Architecture) principles for the whole system has been emphasized by the interviewees.

The central outcomes of the architecture's development [4] are structures for design, execution and monitoring of privacy preserving business processes. The focus is laid on the customer-specific declaration of privacy rules. They are monitored with strategic, operative and tactical KPIs (Key Performance Indicator) with the focus on price, time as well as quality. The quality aspect is verified through test cases development, evaluation and continuous improvement. Through functional and non-functional requirements as well as description of desired conditions, a requirements catalog has been developed, which comprises five architecture-specific criteria. The architecture's functionality is defined through component diagrams and descriptions as well as interface descriptions.

The functional requirements are briefly introduced:

1) Service Selection: The first requirement is the ability of reusing process activities from a (IT) Service Catalog, e.g. as proposed in [26]. It can both accelerate and facilitate the process design. However, it is not to think

about a general service repository in the sense of SOA context such as WSDL (Web Service Description Language) directory, but as repository for structuring and retrieval of customized business process activities, that in BPMS context are known as Work Items.

2) Remote invocation: This criteria comprises interface and functionality for remote process administration and control. For instance, one of the architecture's main components is able to start the process per remote invocation, while another component should be capable of the process' deployment. Moreover, one important feature is the parameter delivering and its acceptance by process during its execution, as well.

3) Process Activity Logging: In enterprise systems, transaction are executed and persisted for compliance as well as for monitoring reasons. The logging of each step during process execution is a challenging task. There is a need for so-called Safe-Points within the process because of real-time process tracking, which is done by parts of the architecture's external components for process monitoring. Safe points save the state of the process instance during process execution. The logging support is required for the process execution engine, as well as for the trigger performance for remote control.

4) Security Provider: The environment of architecture's components should be homogeneous and privacy preserving. Processing with the Single-Sign-On technique can guaranty confidentiality and ease of use for its clients. As a result, one of the requirements on BPMS is the support and integration with OpenID systems.

5) Cloud Readiness: As the architecture's business processing is managed in the cloud, a crucial requirement is the BPMS' support of various features such as support of a CLI (Command Line Interface) for process administration, in order to ease project management. Multitenancy feature support is another requirement because of the necessity of process virtualization. This enables independent execution and design for each particular user or groups of users according to particular roles. It comprises also the possibility to use e.g. customized data sources, or customized persistence configurations.

The trust evaluation index system proposed by [27] was used in order to define non-functional requirements. The three chosen criteria are: reliability, security, and maintainability. The BPM tools will be further evaluated towards meeting the requirements usability and system architecture. Altogether, five non-functional criteria with the following key points to be supported have been defined:

1) Reliability: process monitoring, simulation and design as well as process mining
2) Security: data security and confidentiality as well as process roles and access
3) Maintainability: testability as well as product support and service
4) System Architecture: SOA/Enterprise-components, repository, audit/history logs, human task, workbench
5) Usability: both installation complexity and GUIs (Graphical User Interface) for modeling, process and project administration

Summarizing the first step, there are five aspects determined for the BPMS evaluation that outline the functional requirements.

B. Identifying BPM Tools

The architecture's objectives are being used to derive organization types and use cases in accordance with particular aspects of the decision framework proposed by [28] in order to choose the best fitting BPMS. According to [24], there are some common analytical methods for software tools identification, which comprise market review, rough selection as well as gathering of price offers. For the current paper, the market review and online sources were used. Multiple BPMSs were detected. However, the majority of the BPMSs did not provide the capability of process execution, but only the capability of process design. As a result, common BPMSs are distinguished from BPMN-Engines, which do offer the capability of process execution. Consequently, the list with BPMN engines from [29] was used and extended with the BPEL-Engines from the list of [30] that support BPMN 2.0 language. The list comprised more than 30 BPMN 2.0 engines. Derived from the architecture's setup, the features of BPMN 2.0 Core are determined as essential for the evaluation and selection. Consequently, the list was reduced to seven BPMN 2.0 engines, i.e. Edorasware, Camunda BPM, Imixs Workflow, jBPM, Stardust, W4 BPMN+, and inubit BPM.

By now the BPMN engines have been evaluated based on the BPMN 2.0 Core characteristics. However, reliability and similar aspect such as process monitoring, simulation, and deployment are not part of the BPMN specification [17] but should be considered for evaluation, as well.

C. Selecting BPM Tool

With the aforementioned non-functional requirements, the seven identified BPMS are evaluated. Final results of this assessment are presented in Figure 2. Harvey Balls are chosen as the rating scale. Only jBPM resulted to fully meet all non-functional requirements. However, through interviews with project managers, there was defined an importance of each particular non-functional requirement. The highly weighted are reliability, system architecture as well as security (en-framed bold). Hence, three BPMN engines were chosen to be thoroughly examined concerning the functional requirements in the next step.

The BPM tools are being pre-chosen according to the non-functional requirements. The remaining three are now evaluated concerning the extent to which they meet the functional requirements. The weight of the sub-requirements are presented in the following list:

- Service Selection: Work Item Task (50%), Work Item Repository (50%)

Weight			jBPM	Imixs-Workflow	Stardust	Edorasware	Camunda BPM	inubit BPM
high	I	Reliability	●	●	●	●	◕	◕
high	II	System Architecture	●	●	●	◕	●	●
high	III	Security	●	●	●	◑	◑	◑
low	Iv	Maintainability	●	◕	◕	◕	◕	◕
low	V	Usability	●	●	◑	●	●	◕

Fig. 2. Evaluation of BPMS against the developed requirements.

- Remote Invocation: Remote API for process administration (50%), parameter mapping (50%)
- Process Activity Logging: Signal Events with Remote API (Application Programming Interface) (50%), parameter mapping (50%)
- Security Provider: Support and its full integration (100%)
- Cloud-Readiness: CLI (50%), Virtualizing (50%)

Regarding the executed evaluation, there is only one BPMS fulfilling all functional requirements, i.e. *jBPM*. The tool jBPM is open a source product supporting BPMN 2.0 since 2013. It comprises a huge community and provides a comprehensive and well written documentation. Further, product maturity and integrity can be assumed as it is available in the sixth stable release (currently: 6.5). In the next section, the implementation of the jBPM and the integration with the other existing parts of the collaborative BPaaS architecture is described.

IV. Implementation of BPMS

The following section presents the general implementation of jBPM in the context of a collaborative BPaaS and the implementation of the architecture-specific functional requirements in particular. Further, performance optimization methods for BPMS processes are described. To increase the comprehensibility, some short code snippets are presented.

The integration of privacy evaluation into each activity of the business processes can be achieved by evaluating the privacy at the start of the activity and annotating the data produced by the activity with the appropriate privacy policies. This functionality is not provided by jBPM originally, but it is implemented by creating customized *Work Items* which represent the activities of the business processes. Such Work Items have to be kept extremely flexible. This Work Item is called "Execute_Generic_Task" and is the core element of the privacy extensions added to jBPM.

A. Service Selection

The first functional requirement is the service selection from a service catalog. Two out-of-the-box services are provided by jBPM, which can be used for external communication between the components from within the process. However, there is not much flexibility provided by them, as their data fields and behavior are predefined and cannot be adapted easily. The most adaptive design provides the interface *AbstractWorkItemHandler* that can be used within customized *Work Items*. Its implementation is realized as a plug-in into the process modeling palette. There are two interfaces available: *WorkItemHandler* with only two methods to be implemented and the abstract class of *AbstractWorkItemHandler*, which implements *WorkItemHandler*-interface. Multiple useful methods are provided and are further discussed in the following. The abstract class that takes over *StatefulKnowledgeSession* object as the parameter is responsible for the extended funcationality that has to be integrated. It provides a common way for interacting with the process engine through the following methods: *getProcessIntance(WorkItem wi)*, *getNodeInstance(WorkItem wi)* together with *getSession()*.

First of them provides an access to meta-information about the current process or its parent process, while the second one provides the meta-information about the particular node. Finally, *getSession()*-method gets no parameter and returns the global session that comprises the current session and further various ones being instantiated within the particular project. Further, the above described methods provide also the access from within the node to process variables through the *input/output*-interface. The procedure of registering the customized *Work Item* and to inform the process engine about the customization and registration during process execution, comprises three following steps:

1) *Work Item artifact upload:* After compiling the Work Item as jar-file, it should be uploaded into the process repository which is managed by the maven dependency tool. There are several options available: direct upload through jBPM Workbench with subsequent referencing to it as project dependency in the project management console. The other option is pointing to the deployment ids i.e. groupId, artifactId, and version from the company's external maven repository in the setting.xml file. The Work item will be loaded from the maven repository.
2) *Work Item registration by Runtime Manager:* The AbstractWorkItemHandler has to be registered in the kie-deployment-descriptor.xml file. In code snippet1 line 4 points to the class as well as to its constructor with the ksession as parameter being taken over, where ksession is the global variable registered automatically by jBPM Engine and points to StatefulKnowledgeSession object.
3) *Work Item registration by Work Definition:* WorkDefinitions.wid file provides all important information of Work Item that is being displayed in the design palette as well as its reference to

the Work Item artifact through its name. The list of parameters in code snippet 2 refers to the input elements, that will be taken over by Work Item. This particular Work Item returns no output parameters. Otherwise, they could be listed through results keyword as a list that is similar to the list of input parameters.

Code Snippet 1 Registration of WorkItem by Runtime Manager

```
<work-item-handlers>
  <work-item-handler>
    <resolver>mvel</resolver>
    <identifier>org.example.
        ExecuteGenericTask(ksession)</
        identifier>
    <parameters/>
    <name>CustomWorkItem</name>
  </work-item-handler>
<work-item-handlers>
```

Code Snippet 2 Registration of WorkItem by WorkIDefinitions.wid

```
[ "name" : "ExecuteGenericTask",
  "parameters" : [
    "elementId" : new StringDataType(),
    "ip" : new StringDataType(),
    "email" : new StringDataType(), ],
  "displayName" : "ExecuteGenericTask",
  "icon" : "defaultservicenodeicon.png"
      ]
```

For the WorkItemHandler interface registration the first and the third steps are the same. However, as an interface provides no constructor, it cannot be registered via kie-deployment-descriptor.xml file, but it should be registered in kmodule.xml file, where the session as well as its setups should be provided manually. Having Work Items being uploaded as well as registered in the project, they are displayed in the modeling palette on the left side as shown in fig. 3. Both work items "Get_Initial_Request_Data" and "Service Selection" are instances of "Execute_Generic_Task" being renamed according to their purpose. This is possible because of the generic implementation being written with AbstractWorkItemHandler interface.

B. Remote invocation

The possibility to run the process by remote invocation is provided by the jBPM Remote API (Application Programming Interface). From the multiple use cases which are provided by this interface, there are suitable ones for the project-specific requirements to be used by some of the project's components. Remote process invocation from within the privacy management system taking over the user's IP address and email address as well as remote process deployment by the configurator are crucial interface of the architecture. To realize the first use case, there are two components to be known, namely process definition id as well as deployment name. Code snippet 3 shows the web service call which is necessary to start the process with process definition id "main:project:1.0" and deployment name "project.generic-process". The process definition id is created by stringing together repository, project name, and version delimited by colons, while the deployment name comprises repository and process name which are delimited a dot. While creating the URL it is crucial to take care of data types and automatic type recognition of the application server and framework because of e.g. the IP address which contains dots is not interpreted as a floating point number. This can be ensured by masking critical characters of the values. After request is received and the process is started, the variables are being assigned to the global process variables as shown in code snippet 4. This way their values can be accessed in all Work Items of the whole business process.

Code Snippet 3 The URL for starting the process

```
Http post = new HttpPost("https://<host
    >:8443/jbpm-console/rest/runtime/main:
    project:1.0/process/project.generic-
    process/start?map_ip=user_ip&map_email
    =user_email")
```

Code Snippet 4 Assigning taken over variables to process global ones (kcontext is jBPM's global variable being instantiated by default and points to ProcessContext object)

```
kcontext.setVariable("email_var",
    kcontext.getVariable("email"));
kcontext.setVariable("ip_var", kcontext.
    getVariable("ip"));
```

C. Process Activity Logging

One of the most important project requirements to be implemented is the Safe Points one. It's main objective is to provide process logging after each activity while the process is executed. On the one hand, Signal Events in jBPM are provided via a REST API (Representational State Transfer API) and execution can be resumed programmatically through the BPMS-controller after the log file is read off and all its content is sent to Log-Collector (i.e. the architecture's logging database collecting all operational logs during process execution). On the other hand, as already mentioned, all the logic of the process being packed

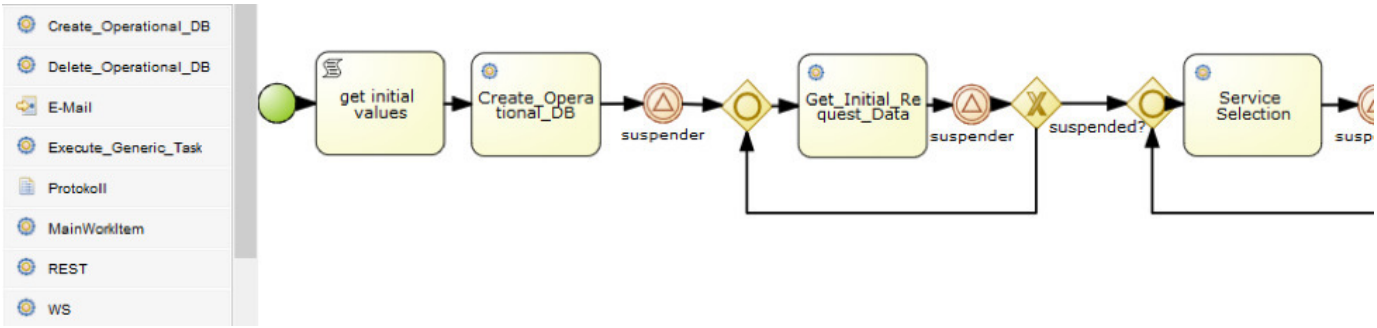

Fig. 3. Business process with custom Work Items being plugged in jBPM Workbench

within the Work Item, each Work Item should control privacy compliance, and, if the privacy rules are violated, the process should be stopped. If the privacy violated could be resolved by the user the process needs to be continued from the node at which the process was stopped. For this purpose XOR-gateways (see Fig. 1) were implemented, which consume boolean values being supplied by Signal Event after a process resume request. If the value is false, the process continues its execution. Otherwise, the Work Item creates a link to a web service which enables the process to be continued manually and sends it to Log Collector. The Cockpit retrieves the link from the Log Collector and presents it to the user indicating the node where the process was suspended. When the user clicks on the link the process is resumed from the activity which failed the privacy evaluation.

D. Security Provider

As discussed earlier an important step in terms of privacy and security of BPaaS can be achieved by securing the cloud environment itself. This can be done by selecting and implementing secure components. E.g. keycloak is an open source identity and access management system, that provides the easy to integrate and secure single sign on mechanism OpenID connect. By providing a minimal configuration, it is possible to provide fully integrated process security with the service provider and user information being known. After keycloak's deployment to the application server and the server adapter (part of the architecture's gateway) being installed, the setup parameters for integration with client and for identity provider can be provided through web-based GUI. Moreover, there is a list of parameters being used during client's integration e.g. type of ssl being required, principle-attribute selection as well as basic authentication enabling, which provides flexible setup of the system. Altogether, keycloak is an easy to implmement tool to provide security integration for jBPM environments which is also very reliable.

E. Cloud-Readiness

Due to the multi-tenancy of clouds the data of each business process have to be kept separated from each other. Following this, the unintentional data transfer between two business processes is avoided. To ensure such a strict separation of the business processes each business process is packaged in an individual project. This project is then deployed to the jBPM server. Then the process can be started independently by each user. This steps of this approach are as follows.

Code Snippet 5 Distinguishing properties in pom.xml to be changed

```
<groupId>#{groupId}</groupdId>
<artifactId>#{artifactId}</artifactId>
<version>#{version}</version>
```

1) Preparing Project Artifacts: Since projects in jBPM are managed in a maven repository, a pom.xml has to be created for each project in order to deploy it to the jBPM server. To simplify the generation of the pom.xml for the processes a template has been generated. The template is copied to the project folder and the process ids, i.e. groupId, artifactId, and version, are filled in automatically. This way each deployment's id is different from each other. The relevant section of the template is depicted in code snippet 5.

Business processes which have been designed in the Configurator are uploaded to an XML database (see sixth criterion's implementation) as xml files. The BPMS provides an API to the Configurator to upload the newly designed process to the database and to send a request to the BPMS-controller containing all relevant variables as described above. The necessary steps to replace the variables in the template and to prepare the deployment of a business process are shown in code snippet 6. The variables are extracted from the http request and are used in line 3 to replace the placeholders in the template pom.xml file. In line 4 the BPMN file for the business process to be deployed is retrieved from the XML database.

2) Process deployment: Having all artifacts being appropriately prepared for project deployment, it is important to check, whether a deployment with the same id already exists. jBPM provides a REST API for both project deployment and un-deployment. Hence, both methods can

Code Snippet 6 Generation of process specific pom.xml and bpmn file

```
Path path = Paths.get(System.getProperty(
    "user.home") + "/standalone/generic-
    project/pomToGenerate.xml");
String pomContent = new String(Files.
    readAllBytes(path));
String pomContentReplaced = pomContent.
    replace("#{groupId}", groupId).replace
    ("#{artifactId}", artifactId).replace(
    "#{version}", version);
String processContent = baseXService.
    getXMLContent(groupId + ":" +
    artifactId + ":" + version + ".xml");
Files.write(Paths.get(System.getProperty(
    "user.home") + "/standalone/generic-
    project/project-to-deploy/src/main/
    resources/generic-process.bpmn2"),
    processContent.getBytes());
```

be used consecutively to guarantee the deployment of the project will execute without any conflicts. Line 1 of code snippet 7 shows the implementation of project's undeployment, while line 5 shows the deployment of the project. Line 3 invokes the maven script in order to compile the project and upload it to the external artifact repository.

Code Snippet 7 Deployment phase

```
restService.trigger(groupId, artifactId,
    version, Trigger.UNDEPLOY) //see
    method's implementation in the code
    snippet 8
...
Runtime.getRuntime().exec("mvn -f" +
    System.getProperty("user.home") + "/
    standalone/generic-project/project-to-
    deploy" + " clean install deploy");
Thread.sleep(30000);
restService.trigger(groupId, artifactId,
    version, Trigger.DEPLOY); //see method
    's implementation in the code snippet
    8
```

F. Process operational data's external storage

As Work Items consume various multiple operational variables, the number of I/O mappings grows linear to the number of consumed variables. Hence, the possibility of errors grows as well. This results in a big performance issue which will be addressed in further research. In order to solve this problem, variables are stored to the external XML database. The following criteria are defined for their selection: REST interface support, GUI provision as well as compliance to standards. The selected application BaseX provides the usage of both XPath and XQuery via REST API. Thus, documents' content can be queried flexibly by the possibility to query either XML nodes or specific values of attributes. Code snippet 9 provides one of the methods used during project implementation to achieve a decision by querying attribute's value. ProcessInstanceId is the id of the current process instance. It aso names the referring table which is created during the process' execution. In order to ensure privacy, the XML file with the all operational data is deleted at the end of each process. The implementation as well as the creation was provided through Work Items. Fig. 4 contains a general view of the business process being designed to meet all requirements. As already mentioned, Work Items provide the activity's full range of functionality. The first activity of each process is not a Work Item, as the first activity is always "Get Initial Values". This activity gathers the initial process input data and stores them in global variables. Those global variables are then consumable by the following tasks.

Code Snippet 8 Implementation of the method in order to un/deploy the project

```
public int trigger(String groupId, String
     artifactId, String version, Trigger
    trigger) {
...
HttpPost post = new HttpPost(hostname + "
    :8443/jbpm-console/rest/deployment/" +
     groupId + ":" + "artifactId + ":" +
    version + "/" + trigger.toString().
    toLowerCase());
//evoke request
//return status code }
```

Code Snippet 9 Retrieving values from XML-DB with BaseX API

```
public Map<String, List<String>>
    getDataByAttribute(String
    processInstanceId, String element,
    String attribute, String value) {
  HttpGet get = new HttpGet(hostname + "
     :8443/" + processInstanceId + "?
     query=//" + element + "[@" +
     attribute + "=" + value + "]");
  //execution
}
```

V. Conclusion

Collaborative BPaaS is an innovative and important approach in order to plan, execute and monitor business

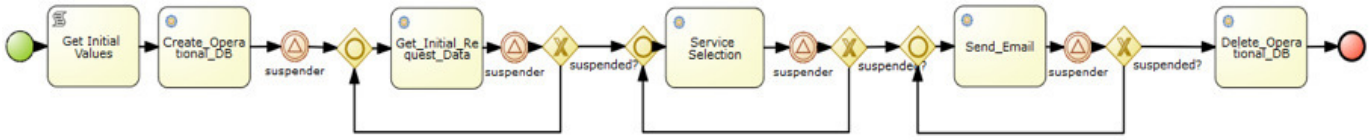

Fig. 4. Evaluation example process.

processes with several involved companies in the cloud. Next to benefits such as increased flexibility and easy reconfiguration, however, the decentralized handling of sensitive data implicates the crucial challenge of companies' individual compliant privacy preservation. In order to meet this challenge, an IT architecture and several of its components have already been developed. This paper presents the implementation of a business process management system as one of the essential components of this architecture. After deriving the requirements for this component, existing BPM tools have been roughly sorted and the remaining one have been evaluated against the requirements. As a result, jBPM is evaluated as the most suitable BPM tool that is able to handle the BPMN 2.0 Core specification and thus, jBPM is implemented into the existing architecture.

Limitations result from the implementation in one exemplary programming language, i.e. Java, as well as from the selection that is influenced by the research induced requirements. Requirements and programming language preference may vary in other contexts of application.

Implication for research is, to the best of our knowledge, the first scientific approach of selecting and implementing a BPMS for cloud-based collaborative business processes in the context of BPaaS. Thus, companies are enabled to collaborate through the cloud while ensuring the privacy preservation of sensible data complying to individual companies' policies. Enabling and realizing the cloud-based collaboration while preserving data privacy, participating companies obtain a decisive comparative advantage.

The system has already been evaluated in several experiments and in interviews with experts from research and practice. Currently, the system is being evaluated in the context of real life use cases within the business environments of several companies. One crucial point for developing the system towards applicability is the optimization of its performance. Starting points for this are the parallelization of privacy evaluations or the parallelization of differing process paths' evaluation.

In summary, the acceptance of cloud computing in general and collaborative BPaaS in particular can be increased by ensuring the preservation of individual customizable data privacy as an important precondition in order to build up trust in the cloud paradigm.

References

[1] P. Mell and T. Grance, "The nist definition of cloud computing," *Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg*, 2011.

[2] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, p. 50, 2008, ISSN: 01464833. DOI: 10.1145/1496091.1496100.

[3] D. Chen and H. Zhao, "Data security and privacy protection issues in cloud computing," in *International Conference on Computer Science and Electronics Engineering (ICCSEE), 2012*, Piscataway, NJ: IEEE, 2012, pp. 647–651, ISBN: 978-0-7695-4647-6. DOI: 10.1109/ICCSEE.2012.193.

[4] B. Schwarzbach, M. Glöckner, A. Schier, M. Robak, and B. Franczyk, "User specific privacy policies for collaborative bpaas on the example of logistics," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2016, pp. 1205–1213.

[5] R. Woitsch and W. Utz, "Business process as a service: Model based business and it cloud alignment as a cloud offering," in *2015 International Conference on Enterprise Systems (ES)*, IEEE, 2015, pp. 121–130, ISBN: 978-1-4673-8005-8. DOI: 10.1109/ES.2015.19.

[6] B. Schwarzbach, M. Glöckner, A. Pirogov, M. M. Röhling, and B. Franczyk, "Secure service interaction for collaborative business processes in the inter-cloud," in *2015 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, IEEE, 2015, pp. 1377–1386. DOI: 10.15439/2015F282.

[7] B. Schwarzbach, A. Pirogov, A. Schier, and B. Franczyk, "Inter-cloud architecture for privacy-preserving collaborative bpaas," *QUIS14*, 2015.

[8] S. Strahringer, Ed., *Business Engineering*, ser. HMD. Heidelberg: Dpunkt-Verl., 2005, vol. 241, ISBN: 9783898643139.

[9] P. Alpar, R. Alt, F. Bensberg, H. L. Grob, P. Weimann, and R. Winter, *Anwendungsorientierte Wirtschaftsinformatik: Strategische Planung, Entwicklung und Nutzung von Informationssystemen*, 7., aktual. u. erw. Aufl. Wiesbaden: Springer Vieweg, 2014, ISBN: 978-3-658-00521-4. DOI: 10.1007/978-3-658-00521-4. [Online]. Available: http://dx.doi.org/10.1007/978-3-658-00521-4.

[10] P. Mertens, F. Bodendorf, W. König, A. Picot, M. Schumann, and T. Hess, *Grundzüge der Wirtschaftsinformatik*, 11. Aufl. 2012, ser. Springer-Lehrbuch. Berlin and Heidelberg: Springer, 2012, ISBN: 978-3642305146. DOI: 10.1007/978-3-642-30515-3. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-30515-3.

[11] H. J. Schmelzer and W. Sesselmann, *Geschäftsprozessmanagement in der Praxis: Kunden zufriedenstellen, Produktivität steigern, Wert erhöhen : [das Standardwerk]*, 8., überarbeitete und erweiterte Auflage. München: Hanser, 2013, ISBN: 978-3446434608.

[12] J. Becker, M. Kugeler, and M. Rosemann, Eds., *Prozessmanagement: Ein Leitfaden zur prozessorientierten Organisationsgestaltung*, Siebte, korrigierte und erweiterte Auflage. Berlin and Heidelberg: Springer Gabler, 2012, ISBN: 978-3642338434.

[13] H. Österle, W. Brenner, and K. Hilbers, *Unternehmensführung und Informationssystem: Der Ansatz des St. Galler Informationssystem-Managements*, ser. Informatik und Unternehmensführung. Stuttgart: Teubner, 1992, ISBN: 978-3-519-12184-8.

[14] F. Bayer and H. Kühn, *Prozessmanagement für Experten: Impulse für aktuelle und wiederkehrende Themen*, ser. SpringerLink. Berlin, Heidelberg and s.l.: Springer Berlin Heidelberg, 2013, ISBN: 978-3642369940. DOI: 10.1007/978-3-642-36995-7. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36995-7.

[15] T. Allweyer, *BPMN - Business Process Modeling Notation: Einführung in den Standard für die Geschäftsprozessmodellierung*. Norderstedt: Books on Demand, 2008, ISBN: 978-3837070040.

[16] J. Freund and B. Rücker, *Praxishandbuch BPMN 2.0*, 4., aktualisierte Aufl. München: Hanser, 2014, ISBN: 978-3446442559. DOI: 10.3139/9783446442924. [Online]. Available: http://dx.doi.org/10.3139/9783446442924.

[17] OMG, *Business process model and notation (bpmn): Version 2.0*, USA, 2011. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/PDF/.

[18] M. H. Hugos and D. Hulitzky, *Business in the cloud: What every business needs to know about cloud computing*. Hoboken, N.J: Wiley, 2011, ISBN: 978-0470616239. [Online]. Available: http://www.eblib.com/patron/FullRecord.aspx?p=624431.

[19] S. Euting, C. Janiesch, R. Fischer, S. Tai, and I. Weber, "Scalable business process execution in the cloud," in *2014 International Conference on Communications and Networking (ComNet)*, Piscataway, NJ: IEEE, 2014, pp. 175–184, ISBN: 978-1-4799-3766-0. DOI: 10.1109/IC2E.2014.13.

[20] B. T. Megersa and W. Zhu, "Cloud-enabled business process management," *International Journal of Computer Theory and Engineering*, vol. 4, no. 5, p. 690, 2012.

[21] G. Herrmann and G. Pernul, "Viewing business-process security from different perspectives," *International Journal of Electronic Commerce*, vol. 3, no. 3, pp. 89–103, 2015, ISSN: 1086-4415. DOI: 10.1080/10864415.1999.11518343.

[22] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012, ISSN: 0167739X. DOI: 10.1016/j.future.2010.12.006.

[23] L. C. Silva, T. Poleto, V. D. H. de Carvalho, and A. P. C. S. Costa, "Selection of a business process management system: An analysis based on a multicriteria problem," in *IEEE International Conference on Systems, Man and Cybernetics (SMC), 2014*, Piscataway, NJ: IEEE, 2014, pp. 295–299, ISBN: 978-1-4799-3840-7. DOI: 10.1109/SMC.2014.6973923.

[24] H. Balzert, *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements-Engineering*, 3. Aufl., ser. Lehrbücher der Informatik. Heidelberg: Spektrum Akad. Verl., 2009, ISBN: 978-3827417053.

[25] C. Rupp, *Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis*, 4., aktualisierte und erw. Aufl. München: Hanser, 2007, ISBN: 3446405097. [Online]. Available: http://deposit.d-nb.de/cgi-bin/dokserv?id=2850705&prov=M&dok_var=1&dok_ext=htm.

[26] M. Glöckner, C. Augenstein, and A. Ludwig, "Metamodel of a logistics service map," in *Business Information Systems*, ser. Lecture Notes in Business Information Processing, W. van der Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, C. Szyperski, W. Abramowicz, and A. Kokkinaki, Eds., vol. 176, Cham: Springer International Publishing, 2014, pp. 185–196, ISBN: 978-3-319-06694-3. DOI: 10.1007/978-3-319-06695-0_16.

[27] C. Li, H. Cui, G. Ma, and Z. Wang, "A bpm software evaluation method," in *Second International Conference on Intelligent System Design and Engineering Application (ISDEA), 2012*, Piscataway, NJ: IEEE, 2012, pp. 1–4, ISBN: 978-1-4577-2120-5. DOI: 10.1109/ISdea.2012.681.

[28] C. Hahn, F. Friedrich, T. J. Winkler, G. Tamm, and K. Petruch, "How to choose the right bpm tool: A maturity-centric decision framework with a case evaluation in the european market," in *EMISA 2012*, ser. GI-Edition lecture notes in informatics proceedings, S. Rinderle-Ma and M. Weske, Eds., Bonn: Ges. für Informatik, 2012, pp. 109–122, ISBN: 9783885796008. [Online]. Available: http://subs.emis.de/LNI/Proceedings/Proceedings206/article6770.html.

[29] Wikipedia, *List of bpel engines*, 2017. [Online]. Available: https://en.wikipedia.org/wiki/List_of_BPEL_engines.

[30] ——, *List of bpmn 2.0 engines*, 2017. [Online]. Available: https://en.wikipedia.org/wiki/List_of_BPMN_2.0_engines.