CLEI ELECTRONIC JOURNAL, VOLUME 21, NUMBER 2, PAPER 6, AUGUST 2018

A Platform Based on Cloud Computing for Executing Collaborative Business Processes

Diego Cocconi
Universidad Tecnológica Nacional (UTN) Facultad Regional San Francisco,
Av. de la Universidad 501, San Francisco, Córdoba, Argentina, 2400
dcocconi@sanfrancisco.utn.edu.ar

Jorge Roa
CIDISI, UTN Facultad Regional Santa Fe,
Lavaisse 610, Santa Fe, Santa Fe province, Argentina, 3000
jroa@frsf.utn.edu.ar

and

Pablo Villarreal
CIDISI / CONICET, UTN Facultad Regional Santa Fe,
Lavaisse 610, Santa Fe, Santa Fe province, Argentina, 3000
pvillarr@frsf.utn.edu.ar

Abstract

With the wide adoption of the Internet, organizations establish collaborative networks to execute Collaborative Business Processes (CBPs). Current approaches to implement Process-Aware Information Systems (PAISs) for executing CBPs have shortcomings: high costs and complexity of IT infrastructure to deploy the PAISs; poor support for organization autonomy, decentralized execution, global view of message exchange, and peer-to-peer interactions; and rigid platforms for generating and deploying PAISs on-demand according to the CBPs agreed in collaborative networks. To overcome these issues, this work proposes a cloud-based platform for the execution of CBPs. The architecture of the platform with the components and interactions between them are described. The architecture enables the generation and deployment on-demand of the PAISs of the organizations required to implement the agreed CBPs, as well as the execution on-demand of these CBPs by fulfilling the above mentioned issues. The platform also provides approaches to deal with elasticity, privacy, and portability concerns.

Keywords: cloud computing, business process, cross-organizational collaboration, process-aware information system.

1 Introduction

With the wide adoption of the Internet by organizations, new markets, and economic conditions, organizations tend to establish integration, cooperation, and collaboration relationships, resulting in new forms of *collaborative networks* [1]. A collaborative network consists of autonomous, geographically distributed, and heterogeneous organizations that collaborate to achieve common goals [2]. Collaborative networks contribute significantly to enhance performance of Small and Medium Enterprises (SMEs) [3].

In a collaborative network the integration and collaboration among the organizations are established and carried out through *Collaborative Business Processes* (CBPs) [4]. A CBP (also called *process choreography* [5][6]) specifies the global view of the control flow of interactions (the message exchange) between roles that organizations perform to achieve a common business goal, and serves as a contractual basis for the *cross-organizational collaboration* involved [7]. The implementation of collaborative networks (and collaborations) requires organizations to have the appropriate technology to execute the CBPs that the organizations agreed to carry out in these networks. The execution of CBPs requires to deal with: (1) *organization autonomy*; (2) *decentralization*; (3) *global view of message exchange*; and (4) *peer-to-peer interactions*.

Platforms for CPBs based on Internet technologies like *Web services* require each organization use its own resources and infrastructure (hardware, software, network, etc.) [7][8] to develop, implement and maintain *Process-Aware Information Systems* (*PAISs*) required by the organization to execute their *Integration Business Processes* (IBPs) [7]. An IBP (also known as *private process* [5] or *orchestration process* [6]) defines the public and private activities an organization has to perform to fulfill the message exchange defined in a CBP. This increases complexity and costs for the organizations. Even though big companies can deploy these kind of solutions, the above aspects have a more negative impact on SMEs, governments of small cities and communities, and healthcare public or private institutions [9]. So it is important to use technologies that can make it feasible for organizations like these the implementation of collaborative networks and the execution of CBPs.

Furthermore, existing proposals do not focus on dynamic aspects of cross-organizational collaborations [10], so they do not provide services for allowing organizations to *generate*, *deploy*, and *enact* (or *execute*) PAISs *on-demand*, accordingly to the CBPs that the organizations agree to carry out. These kind of services would result in more agile collaborations, allowing to set up a collaboration at any moment and enacting more fluidly the involved processes.

Exploiting the benefits of *cloud computing* technologies for the execution of CBPs appears to be suitable to solve the shortcomings of: high costs and complexity of IT infrastructure required to implement PAISs; fulfillment of the requirements of CBPs above mentioned; and rigidity of platforms for PAISs that do not enable organizations to generate, deploy, and enact PAISs on-demand, accordingly to the CBPs agreed in collaborative networks.

Therefore, this work proposes a platform based on cloud computing to offer on-demand services for the execution of CBPs in collaborative networks. The platform allows organizations to: (1) reduce costs and complexity by hiding the required infrastructure required for the PAISs; (2) create and manage collaborations in an agile or dynamic way (on-demand), i.e. by generating and deploying the IBPs and PAISs required to implement the agreed CBPs, and executing and monitoring these CBPs on-demand; (3) fulfill the requirements of CBPs management for all the offered services. The proposed architecture of the cloud-based platform provides a deployment model based on a *public cloud* and *private clouds*. The architecture is based on a three-tier model which consists of *persistence*, *service* and *application layers*. Components of the service layer are those that provides the main functionalities to support the cloud services offered by the platform. Some of these components are based on methods, and tools proposed in previous works [4][7][8].

The architecture is also oriented to fulfill non-functional requirements of cloud computing services, in particular *elasticity*, *privacy*, and *portability* are of major importance. Elasticity is important for improving performance when several IBPs and PAISs are enacted. It is also important to deal with privacy issues related to sensitive information shared in a collaboration, or internal information managed in IBPs, because organizations have to rely in the platform for enacting their processes, which can handle sensitive data. Finally, portability must be considered for conceiving a platform independent of the cloud provider.

This work complements a previous work [11] by adding: a detailed description of main the components of the platform, a proposal of an implementation schema that enables the deployment of the platform in IaaS environments from different IaaS providers/platforms, and a case study that allows showing the applicability of the platform.

The work is organized as follows. Section 2 presents a background related to cloud computing and CBPs. Section 3 summarizes related work. Section 4 describes the defined architecture for the platform and the main functionalities. Section 5 describes a case study with the purpose of validating the platform and showing the use of the proposed cloud-based services. Finally, section 6 presents conclusions and future works.

2 Background

2.1 Cloud Computing

Cloud computing is a new paradigm for creating distributed systems based on the Internet [12]. From a business point of view, cloud computing could be seen as a model for delivering *on-demand* services, where shared resources and applications are provided through the Internet. This model allows organizations pay for resources or applications only when they use them ("*pay-per-use*") instead of facing the considerable costs of procurement and maintenance of a hardware and software infrastructure, and software licensing in consequence [13].

There are three *service models* for Cloud computing: *SaaS* (*Software-as-a-Service*), *PaaS* (*Platform-as-a-Service*), and *IaaS* (*Infrastructure-as-a-Service*) [10]. In the SaaS model, applications are offered as services, accessed through the Internet on-demand by users. This model is mature today and there are several cloud applications available [9]. The PaaS model offers development services for building cloud applications. IaaS model implies the provisioning of hardware resources via *virtualization* [13]. Cloud computing also has different *deployment models*, such as *private cloud*, *community cloud*, *public cloud*, and *hybrid cloud*. In a private cloud, infrastructure is operated solely by one organization and managed by the organization or a third-party. In a community cloud several organizations jointly construct and share the same cloud infrastructure, which could be hosted by a third-party or by one of the organizations. In a public cloud, service provider has the full ownership of the cloud architecture with its own policy, value, profit, costing, and charging model. Finally, the hybrid cloud is a combination of private, community, or public clouds [14].

Independently of the cloud service model, there are some features to be considered by a cloud solution. In this context, *elasticity* (*on-demand scaling*), also known as *redeploying* or *dynamic provisioning* of applications, has become one of the most important issues. Elasticity enables real-time acquisition/release of computing resources to scale the applications up and down in order to meet their run-time requirements or demands, while letting application owners pay only for the resources used [15]. Besides, elasticity is strongly linked to the concept of *multi-tenancy*; multi-tenancy is defined as the capability of a single software instance to provide its service to several parties simultaneously without degradation, emulating multiple application instances [16][17].

Within cloud environments, expectation of *privacy* protection exists everywhere. Privacy refers to sensitive information about one person or a group that is expected to be hidden from others (e.g., identity, address, health, and hobbies). When users access cloud data and use cloud services, it is necessary to preserve their privacy, which generally is achieved by utilizing various encryption techniques [18].

In an ideal situation, cloud users would be able to easily switch providers to take advantage of a better service or price. However, the reality of cloud computing today is somewhat different to this outlook. In particular, *vendor lock-in* is a significant barrier that must be overcome. Vendor lock-in refers to the dependency created between the cloud user and cloud provider since the cloud user deploys their software on the provider's cloud platform or infrastructure. This dependency is created due to the heterogeneity of the services offered by different cloud providers [19]. Formally, *portability* is the ability to move software assets among different runtime environments without having to rewrite them partly or fully. If the cloud portability is achieved, data, application, or service components can be moved and reused between different cloud providers regardless of the operating system, storage, or Application Programming Interface (API), avoiding the vendor lock-in problem [20].

2.2 Collaborative Business Processes

To carry out collaborations, organizations integrate their business processes, agree on common business goals, coordinate their actions, and exchange information by defining and executing CBPs [21]. The implementation of collaborations implies that organizations can carry out the stages of the BPM lifecycle [5] to the CBPs involved. During the *analysis* and *design stages*, organizations have to define not only CBPs but also their internal business processes (IBPs) that model the private behavior of organizations and support the interactions and roles they perform in the collaboration. To deal with organizational autonomy issues, CBPs are defined as abstract processes. This means they are not directly executable by a centralized engine. Instead, they are enacted by means of the IBPs of each organization in a decentralized way [7].

The next stage is *implementation*, which consists in the development, configuration, and deployment of PAISs required for each organization to execute their IBPs. Cross-organizational collaborations rely on the capability of

PAISs to interoperate for managing CBPs. This implies that each organization has to implement a PAIS to enable the execution of its own IBPs and interact with the other PAISs to achieve the message exchange agreed in CBPs [7].

The *execution* stage consists in the enactment of the CBPs by means of the enactment of IBP instances by the PAISs of each organization, to execute the private activities organizations need to carry out, as well as the public activities related with the message exchange with each other. A decentralized management brings an additional challenge for CBP monitoring: to know with certainty the execution state of CBPs and to deliver this information to the (correct) interested organizations.

There are languages that emerged to support the modelling of CBPs, such as BPMN [6] and UP-ColBPIP [4][22]. These languages enable the definition of technology-independent models of CBPs. The purpose is to describe CBPs at a high level of abstraction from a conceptual point of view, providing concepts closely related to the business and organizational domains, which are suitable for both business and technological people.

Moreover, collaborations may be defined from different *perspectives*, i.e. considering a *global* picture of the control flow of interactions, or considering a *local* picture of the public and/or private activities of each organization together with the interaction points among them. The correspondence between these perspectives and the types of processes is shown in Figure 1.

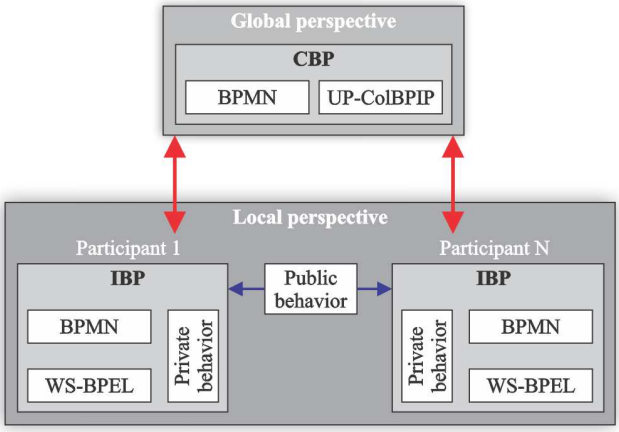


Figure 1: Global and local perspectives of cross-organizational collaborations.

The global perspective describes the global and public behavior of a cross-organizational collaboration and the responsibilities of participants [4]. This behavior is represented as a single control flow that defines the order in which interactions (exchange of messages) between participants take place. This means that interactions are defined as if they were seen by an observer who can only see the public and externally visible behavior of each participant. Figure 1 shows examples of technology-independent languages that are oriented to represent this perspective, such as BPMN (choreography diagrams) and UP-ColBPIP (interaction protocols).

The local perspective describes the behavior of a collaboration based on the activities of each participant. In this perspective, a collaboration is described as a composition of business processes that define the independent behavior of each participant and the interactions between them. Each participant has its own control flow and interaction points. In the local perspective, two types of business processes for each participant can be defined: a public behavior (*interface process*) and a private behavior (Figure 1). An interface process describes the public and externally visible behavior of a participant in terms of the activities that support the receiving and sending of messages with each other [4][5][23]. On the contrary, the private behavior describes both the public and private actions required to support the role a participant performs in the collaboration. Unlike interface processes, the private behavior adds the internal activities and events required to support the internal business logic of a participant. BPMN enables the definition of both kind of processes in an independent way of the implementation technology. WS-BPEL enables the definition of both of them based on the Web Services technology.

2.3 Cloud Computing for CBP Management

Related to cloud computing and the BPM domain, a new service model has emerged: *Business Process as a Service* (*BPaaS*). BPaaS is a new category of cloud service, which can be defined as the delivery of *BPO* (*Business Process Outsourcing*) services that are sourced from the cloud and constructed for multi-tenancy. BPO services are non-computational services such as logistics, supply chain, or IT delivery services, that are based on the provisioning of business processes as services, procuring partial or full business process outsourcing. Like computational services, their execution is regulated by Service Level Agreements (SLAs) and supported by specific software, usually PAISs. As a cloud service, the BPaaS model is accessed via Internet-based technologies [24]. Then, by means of BPaaS, processes can be executed and accessed over the Internet [25].

The application of BPaaS for CBP execution would significantly reduce costs and complexity to organizations, avoiding them to use their own infrastructure for implementing the PAISs required for the IBPs that have to be executed. However, agile collaborative networks require new cloud services that enable the creation and execution management of CBPs in a dynamic way (on-demand), i.e. by generating and deploying the IBPs and PAISs of the organizations required to implement the CBPs that are agreed by them at any moment, and executing and monitoring these CBPs on-demand. Therefore, the aim of this work is to propose a new kind of PaaS model for CBPs that enables the definition, generation and deployment of the IBPs and PAISs, in addition to the support of the execution of the CBPs by means of the IBPs.

Furthermore, applying cloud computing for CBP management implies the development of a solution that takes into account the two perspectives for collaborations. Global perspective to take into account requirements of CBP execution: organization autonomy; decentralized execution; global view of message exchange; and peer-to-peer interactions. Local perspective considers the execution of IBPs, so fulfillment of requirements of autonomy and privacy have to be contemplated in this case.

3 Related Work

Existing approaches for offering BPM in cloud environments mainly focus on fulfilling elasticity issues [26][27][28][29]. In [26] the solution seems to be very satisfying because it considers decentralization of elasticity control and the idea of generic strategies for dealing with elasticity. In [29] it is highlighted the fact that SMEs are falling behind in cloud usage due to missing IT-competence and hence losing the ability to efficiently adapt their IT to their business needs, and the authors propose a PaaS architecture to cope with this issue. Authors of [30] propose a PaaS model for cloud applications implemented in terms of active components to accomplish non-functional requisites. This implies the development of a platform that manages elasticity in terms of these components and the non-functional requisites.

Other works focus on security and privacy in the cloud. In [31] authors propose an anonymization-based approach to preserve client business activity while sharing process fragments between organizations on the cloud. This is because the cloud model gives the opportunity to mash up and compose services from a variety of cloud providers to create what is known as a *cloud syndication*. In [32] the author exposes that although outsourcing (the execution of business processes) harbors an economic potential, cloud consumers lose control over their data and executions. Moreover, both external and internal auditing pose a general challenge for their acceptance, so he reviews the role of remote auditing as a mean to address this issue, reporting on the main properties to be considered: compliance and security for BPaaS (authorization, separation of duty, binding of duties, delegation, conflict of interest, etc.). Although these works provide interesting approaches for privacy and security, they do not focus on privacy and security in CBPs that are executed in a decentralized way in a same or different clouds.

There are few proposals that support cloud services for CBPs, such as [33][34][35]. In [34] authors present a SOA (Service Oriented Architecture) architecture based on an ESB (Enterprise Service Bus) for supporting collaborative processes. The general approach of the project is based on three levels: (1) business level (the use of a generic metamodel for collaborations); (2) technical level; and (3) agility level (management of evolutions and changes on situational models). However, the architecture does not offer a clear vision of a cloud service model that could be used for supporting it. It is focused on offering interoperability (not portability) and agility not in the sense of dynamism for supporting CBPs on-demand, but in a system that should remain well adapted to the potentially changing needs of collaborations instead (contexts that change, networks of partners that change, etc.).

In [35] authors argue that multi-tenancy for BPM systems is necessary for effective BPaaS, but it is technically hard to realize, because existing business process design methodologies lack coherent plans for data design –i.e. business data for the process logic, business process models, execution states, correlations among business process instances, and resources and their states. Without coherent data design, current BPM systems handle and manage data in ad hoc manners, so the business process execution is scattered across databases, auxiliary data stores managed by the BPM systems, and even in files. They think that a unique and shared business process engine connected to a unique datastore it is an improved situation, with both the data and process definition being provided to the process engine when it needs to schedule tasks. But this makes the process engine more complex (because it has to lead with several concurrent process instances) and difficult for scaling it when the workload increases. All of these proposals have still important shortcomings for collaborative networks: (1) they provide a centralized approach for CBPs being their execution driven by one organization, and do not deal with autonomous process execution; and (2) they offer SaaS models to execute processes as services, but not a PaaS model –i.e. not providing the possibility for generating and customizing the technological solutions required to execute new CBPs that organizations agree to be carried out on-demand for their needs.

Finally, outside the area of cloud computing, there are also software agent-based platforms proposed for executing CBPs [36][8]. In particular, in [8] a platform that deals with the issues of dynamic collaborations is proposed. However, these proposals require organizations to deploy PAISs in their own private infrastructures, which implies more complexity, costs, and poor agility for managing collaborations. This results in a more difficult adoption of these solutions by the organizations that are interested in the implementation of collaborative networks.

4 Cloud-based Platform for CBPs

This section describes the proposed platform based on cloud computing for executing CBPs. The aim of this platform is to provide a PaaS model with services that support the definition and management of collaborative networks and the implementation, execution and monitoring of CBPs. First, the offered services by the platform are described through a simple scenario. Next, a conceptual architecture of the platform is presented and their components and interactions between them are described. Finally, implementation details are given.

4.1 Cloud Services of the Platform

The platform provides services for: (1) defining collaborations along with the CBPs to execute; (2) implementing CBPs by means of inter-operable and consistent IBPs; (3) executing CBPs through the enactment of IBPs by PAISs of each organization in an autonomous way –so interactions among the organizations' PAISs are carried out in a decentralized way–; (4) monitoring the execution state of CBPs.

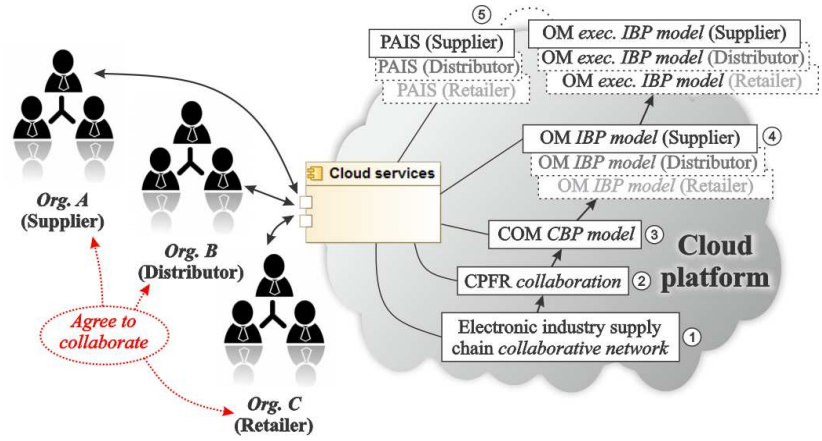
To describe the services and functionalities of the proposed platform, an example of a collaboration scenario from the domain of supply chain of the electronic industry is used (see Figure 2), where three organizations (*Org. A*, *Org. B*, and *Org. C*) performing the roles of *Supplier*, *Distributor*, and *Retailer* respectively, want to implement a well-known collaborative model called *Collaborative Planning, Forecasting, and Replenishment* (*CPFR*) [37]. As part of this collaborative model, organizations require to implement and execute the CBP called *Collaborative Order Management* (*COM*) [38].

4.1.1 Cloud Services for the Implementation of CBPs

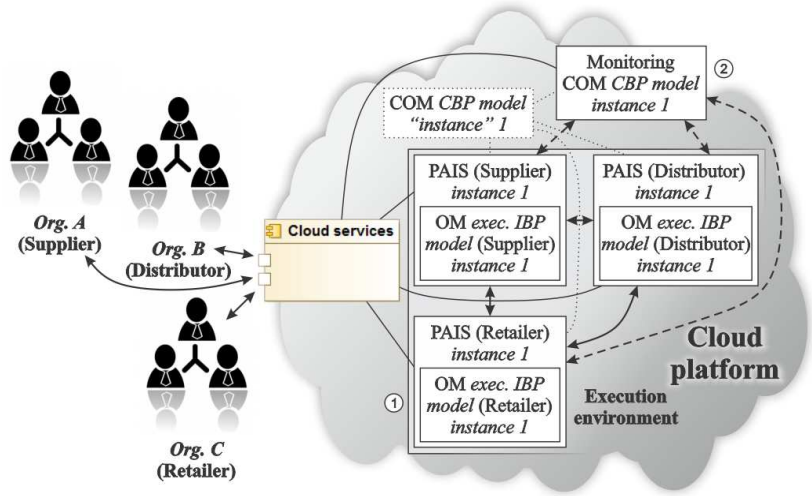
Figure 2a shows the platform provides *cloud services* for the *implementation* of CBPs. First, organizations (Supplier, Distributor, and Retailer) are registered in the *cloud platform* and grouped in a *collaborative network* called *Electronic industry supply chain* (step 1). Then, organizations agree on a new *collaboration* named CPFR, defining goals and agreements to collaborate (step 2).

Afterwards, organizations can agree on the CBPs to be executed in the collaboration. The platform is fully integrated with a repository for collaborations and CBP models that was developed in a previous work [7], so by means of cloud services, organizations can select, update, and propose a *CBP model* or to create and upload a new one according to their needs, and share it as part of the collaboration. In the example scenario, organizations agree on to manage the CBP named Collaborative Order Management (COM) (step 3). In order to guarantee correctness of CBPs

behavior, the platform also provides a service to perform the verification of CBP models by using the method proposed in [39].



(a) Implementation of CBPs.



(b) Execution and monitoring of CBPs.

Figure 2: Cloud services of the platform for CBPs.

Once a CBP model is agreed by organizations, the platform provides cloud services to generate the IBPs that each organization needs for the CBP execution. To support this, the platform implements a model-driven method and tool [23] for generating the *IBP models* for each organization from the agreed CBP model and save them in private repositories, each one accessible only by the owner organization. In the example scenario, an IBP model named *Order Management* is generated for each one: Supplier, Distributor, and Retailer (step 4). Each IBP model contains the activities required to execute the collaboration from the point of view of the corresponding organization, but it is still an incomplete model that has to be configured with execution details. In order to achieve an executable implementation of the CPB, each organization has to complete its IBP model with its private behavior, data, and resources necessary to carry out the activities. At this step, organizations can also use verification services [40] to guarantee their correctness. Then each organization can make use of cloud services to configure its IBP model with execution details, to become an *executable IBP model*, and to generate the *PAIS* that implements this executable process. The generated PAISs and

executable IBP models are also stored in a private repository for each organization, so organizations can manage their own PAISs to carry out collaborations. In the example scenario, PAISs are generated for the Supplier, Distributor, and Retailer, configured with the corresponding executable IBP to be enacted by the organizations (step 5)

4.1.2 Cloud Services for the Execution and Monitoring of CBPs

Once all PAISs are generated and saved, organizations can make use of cloud services to support the execution of CBPs, which is carried out by means of the executable IBP models and the deployed PAISs of each organization. Thus, the platform enables the execution of CBPs through the distributed (autonomous) execution of IBP instances supported by the PAISs (Figure 2b). This way, organizations are able to exchange information with each other in a peer-to-peer way by means of the PAISs, supporting a decentralized execution of CBPs.

Hence, to initiate the CBP execution, the platform provides services to make the deployment of the PAIS of each organization into a proper *execution environment* (creating a PAIS instance in consequence) which enacts an instance of the corresponding IBP model. In the scenario, the PAIS of the Supplier is deployed and it instantiates a process instance of its IBP model. In a similar way, the PAISs of the Distributor and Retailer are deployed to instantiate the rest of the IBPs required to execute the CBP (step 1).

With all PAISs deployed and IBPs already instantiated, the *monitoring service* is started (step 2). This service enables organizations to be aware of the global state of the collaboration and, in particular, of the COM collaborative process. This allows the continuous evaluation of the fulfillment of the established agreements, or to predict the quality of the collaboration with a given partner in the future [41].

4.2 Architecture of the Platform

A general view of the architecture is shown in Figure 3. The platform is divided in a *public cloud* and *private clouds*. A general view of the architecture is shown in Figure 3, where is represented the public cloud above and a private cloud below (more private clouds could be supported, but only one is shown for simplicity). The public cloud contains the services for managing the *collaborative networks*, *collaborations*, and *CBP models*, i.e. all the information shared by all organizations, which is arranged in a *global context*. The public cloud also contains the services for managing the *organization context* for each organization. All the private information of each organization is preserved to be exposed to others via independent *organization contexts*, reachable only by the holder organization. Due to organizations may not want to participate in collaborations by using a public service to manage sensitive information such as the involved in their private IBP models, the platform allows organizations to make use of private clouds interacting with the public cloud. This provides more autonomy to the organizations and fulfills the requirement that some of them would have for maintaining and preserving their information and activities in private.

Hence, the platform can be deployed in a public cloud and in private clouds, all together communicated via a *message broker*. By using the public cloud, all information and processes of an organization are stored and managed over the IT infrastructure of a service provider. Instead, the use of a private cloud by an organization requires the deployment of the platform in its own IT infrastructure or in a third-party service provider selected by the organization. Anyway, the services related to the management of the organization context, i.e. the services for IBPs and PAISs, are the same in both public and private clouds, as can be seen in Figure 3.

The architecture of the platform follows a three-tier model which consists of *persistence*, *service* and *application layers*. Persistence layer groups all the repositories that the platform needs for offering the cloud services. A *CBP model repository* for managing collaborative networks, cross-organizational collaborations, and their CBP models is present only in the public cloud and can be accessed by all organizations (it belongs to the global context). *IBP model repositories* and *PAIS repositories* are present in both clouds and each of these repositories is exclusive for the owner organization. IBP model repositories are used for maintaining IBP models together with templates that organizations can define. PAIS repositories store the code of the PAISs for each organization and the executable specifications of the IBP models.

Service layer provides the components that support all cloud services offered by the platform. For supporting the definition of collaborative networks and the implementation of CBPs, service layer consists of three main components: *collaboration manager*, *IBP manager*, and *PAIS manager*. Collaboration manager handles aspects related to the management of collaborative networks and CBP models. This component provides operations to add/update/remove/get collaborative networks and cross-organizational collaborations, as well as the organizations that

participate in them and the roles they fulfill. Also it provides operations to access the CBP model repository and manage the model versions of CBPs in each defined collaboration. CBP models are based on the UP-ColBPIP language [22]. However, BPMN choreography diagrams also could be used to describe CBPs and they can also be stored in this repository. One of the reasons of using the UP-ColBIP is the possibility to reuse several methods and tools proposed in previous works to verify and transform CBPs.

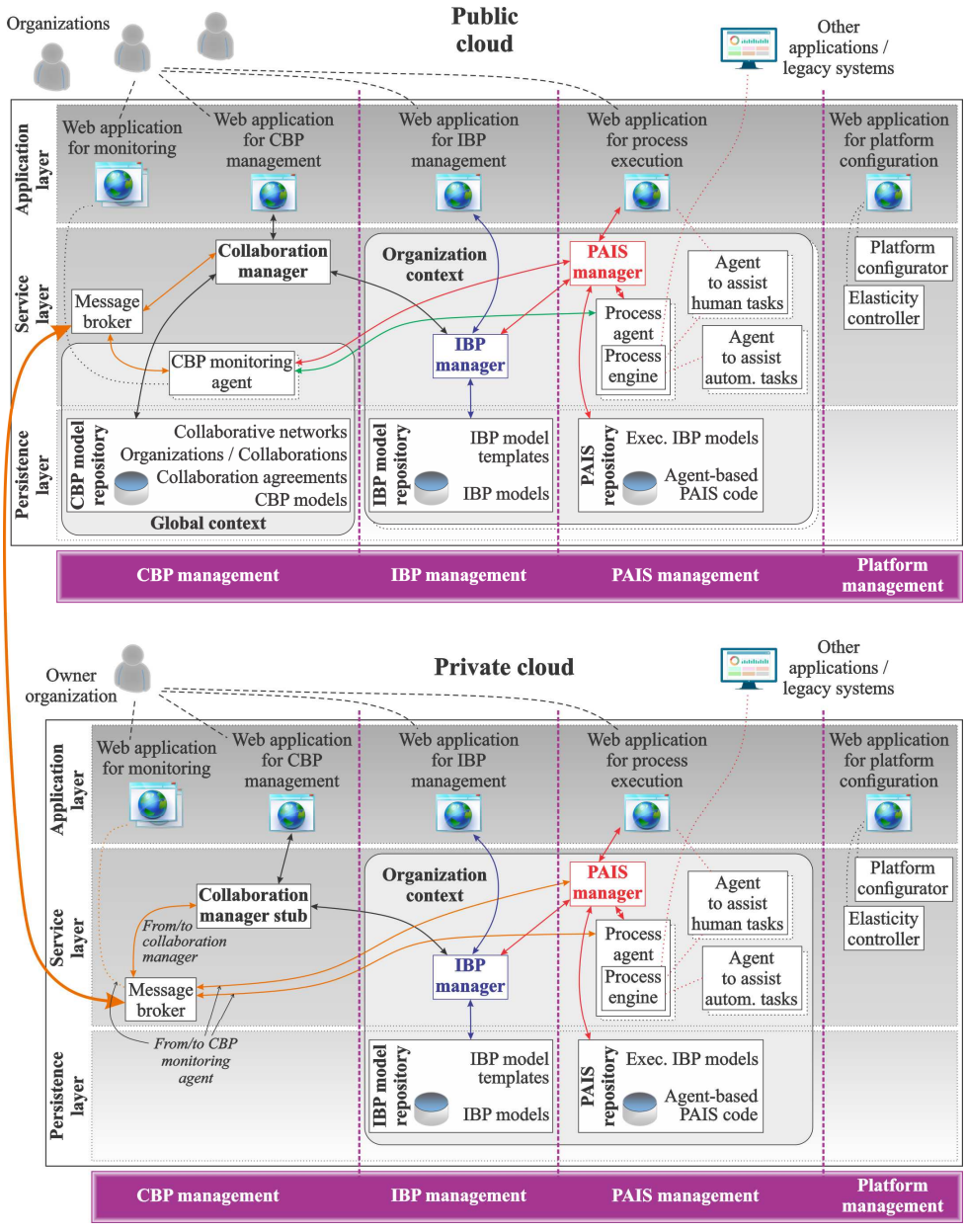


Figure 3: Architecture of the cloud-based platform for CBP management.

The collaboration manager also provides an operation that implements the method proposed in [39] to determine if the behaviour defined in a CBP model (expressed with the UP-ColBPIP language) is error-free (i.e. if it does not have

deadlocks, livelocks or lacks of synchronization), when it is added to a collaboration. If errors were found, the addition of the CBP model is cancelled. Else, the collaboration manager communicates with the IBP manager for generating the IBP models for each organization.

Collaboration manager is a unique entity, resident in the public cloud only. But some type of facade is necessary in private clouds to emulate the provided services and enables the communication with the public cloud regarding the services for collaboration management. Organizations that makes use of private clouds must not distinguish any difference using local services respect to the ones offered in the public cloud. So, a collaboration manager stub in each private cloud becomes the side of collaboration manager in each one of them, replicating the behavior of the collaboration manager for organizations relying in this deployment model. Communication between the collaboration manager and the collaboration manager stubs is realized via the message broker.

An IBP manager is present on each organization context and manages the IBP models of each organization, saving/retrieving these models from the IBP repository. This component provides the operations that implements the MDA-based method proposed in previous work [21] to generate a IBP model of an organization from a CBP model. This method provides a model-to-model transformation process that takes as inputs a CBP model and a target organization role, and automatically generates as output an IBP template model that contains the public and private activities and control flow logic that an organization has to implement for performing the target role in the CBP. The input CBP model is based on the UP-ColBPIP language [22], and the output model is a private process based on the BPMN language [6]. These languages allow representing platform independent process models at a conceptual level. By using the UP-ColBPIP language the behavior of CBPs is modelled through interaction protocols. This method guarantees the generated IBP model is consistent with the input CBP model, and hence interoperable with the IBP models of the other organizations involved in the CBP.

A PAIS manager also is present on each organization context and manages the PAISs of each organization, i.e. to save/retrieve its implementation code from the PAIS repository. The aim of the platform is the PAISs of the organizations can be executed autonomously in the cloud and interact and share information in a peer-to-peer way. To fulfill these features, it is proposed to implement the PAISs in terms of software agents.

In this platform a PAIS of an organization is implemented as a software agent, which is named *process agent*. A process agent contains an embedded process engine and an executable IBP model. Thus, a process agent represents the PAIS of an organization that is able to enact and execute an IBP of the organization. PAIS manager is responsible to generate the code of the process agents. This is carried out by reusing the model-driven method proposed in [8], which allows generating the code of the agent and a template of the executable IBP model from a conceptual IBP model. The PAIS manager gets an IBP model from the IBP process repository, generates the process agent code and the executable IBP model and store both in the PAISs repository. To execute an IBP as part of a CBP execution, the PAIS manager is responsible of the instantiation of the corresponding process agent, which also implies to load the executable IBP model into the embedded process engine of the agent to enact the instance of the IBP model.

PAIS manager also instantiates *CBP monitoring agents* responsible for monitoring the execution state of the CBPs being executed. This state allows to determine if each organization involved in the collaboration has completed its part, who is waiting for some interaction, and the global view of the CBP execution. Due to CBP monitoring agents only reside in public cloud, they communicate with PAIS managers and process agents in private clouds via the message broker.

Besides process agents and CBP monitoring agents, there are other *auxiliary agents* in the service layer. One kind of auxiliary agent is the *agent to assist human tasks* of IBPs, which interacts with the organization to show up the proper forms to complete information by a human user. Other auxiliary agent is the *agent to assist automated tasks* of IBPs, which interacts with external programs or legacy systems, or that could require to invoke programs implementing decisions based on automated rules.

The application layer refers to the Web applications that offer a front-end by means the organizations make use of the services of the platform. These Web applications invoke the APIs offered by components of service layer (collaboration manager, IBP manager, and PAIS manager). The *Web application for CBP management* allows accessing to the services offered by collaboration manager. The *Web application for IBP management* interacts with IBP manager for accessing to the IBP model repository. The rest of the Web applications mainly interact with the software agents, supporting the execution and monitoring of CBPs. The *Web application for monitoring* shows the information provided by the CBP monitoring agents. As these agents reside only in public cloud, the application in private clouds must communicate with them via the message broker. *Web application for process execution* allows

configuring the IBP model generated once the CBP model has been agreed, to become it the fully executable model –enacted by the process agent. Executable IBP models depend of characteristic details of the execution environment –in this case, the process engine embedded into the process agents–, so this application is related to the PAIS manager. Web application for process execution also shows the forms to complete information by a human user when an IBP model is in execution by a process agent, so this application is linked to the agents to assist human tasks. The Web application for CBP management and the Web application for IBP management are reused from the distributed process model repository proposed in [7]. The rest of the Web applications are new applications required by this cloud platform.

Besides the classical tier arrangement of the platform architecture, it is particular useful a transversal view, as vertical divisions show in Figure 3. These divisions group components according to their responsibilities and functions. Collaboration manager, CBP monitoring agents, and CBP model repository are responsible for *CBP management*. These components allow managing CBP models defined with the UP-ColBPIP language. If it were decided to use another language, as BPMN, it is only necessary to change the services in charge to transform CBP models to IBP models. If it were also necessary to choose another monitoring method –i.e. not to use software agents– it is only necessary to change the monitoring component.

In the same way, IBP manager and IBP model repository deal with models of private processes (*IBP management*), using BPMN as a modelling language for their representation.

PAIS manager, PAIS repository, and process agents (and the rest of the software agents) perform the PAIS management. Even though it is planned a PAIS execution through software agents, it could be done using other mechanisms. Only PAIS manager and PAIS repository should be adapted in that case. For the executable IBP models, it is used as process specification language an executable version of BPMN used by the *Jadex Processes framework* [42]. Jadex Processes framework provides a process engine that is able to enact and execute business processes defined and configured with this executable version of BPMN. The Jadex process engine can be embedded into agents to carry out the behavior of the agents based on a process model. However, if it were decided not to use software agents at all, any other process engine could be used as well, such as one supporting WS-BPEL as process specification language.

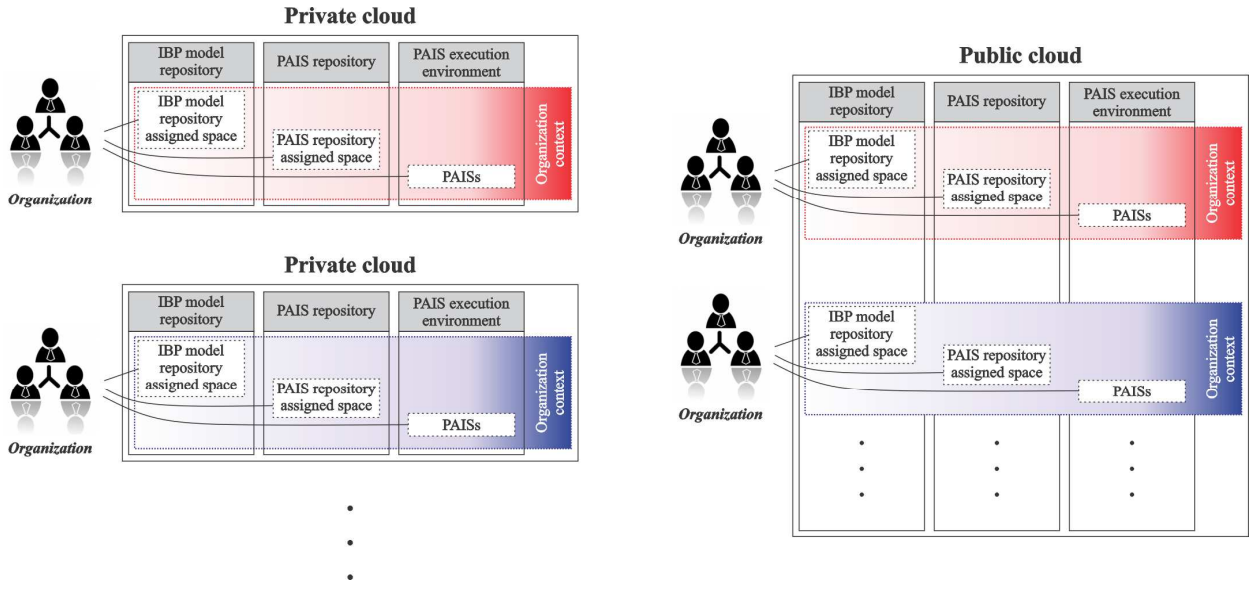
In addition to the components that supports the cloud services offered by the platform, there are other components that support the non-functional requirements of elasticity, portability and privacy.

Elasticity (and therefore multi-tenancy) must be provided from the point of view of repositories, agent execution environments, and Web applications. Multi-tenancy is reflected to provide services without degrading the performance of the platform. So, an *elasticity controller* is considered in both clouds, which has to deal with proper metrics and react in consequence scaling up (or down) instances of the different aforementioned components.

Portability (and vendor lock-in in consequence) is another important issue, considering that the platform is conceived to be independent of the cloud provider or cloud platform. Besides the public cloud, this is particularly useful for private cloud users, because they can freely decide the cloud provider or even deploy their private cloud using their own infrastructure. To help the process of setting up the platform, taking into account portability, a *platform configurator* component is also available.

Finally, privacy is considered at two levels: (1) by means of the private cloud deployment model, the organization context is isolated on an cloud owned by the organization, allowing organizations who prefer this model to interact with the public cloud without exchanging sensitive information; and (2) by means of the organization contexts in public cloud, which enable a logical contained space owned by one organization, where there are located its IBP models, PAIS code, and instances of its software agents. This defines two extremely opposite situations, as can be seen in Figure 4.

The left part of the figure shows a situation where all organizations chose a private cloud deployment model; IBP model repository, PAIS code, and execution environments (for software agents) are not used in public cloud though. The opposite situation, shown in right part of Figure 4, describes the case when all organizations decide to use the public service. So, IBP and PAIS repositories, and execution environments in public cloud are partitioned for supporting all the organization contexts necessary. Combinations between these two extremes are also possible.



(a) All organizations using private clouds.

(b) All organizations using the public cloud.

Figure 4: Organization contexts.

4.3 Components of the Platform

Table 1 summarizes the main components of the platform. For each component there is a brief description of its functionalities, where the component is located (in public or private clouds), and the accessibility level for organizations.

Table 1: Components of the cloud-based platform for CBP management.

	Component	Description	Location	Accessibility
Repositories	CBP model repository	Global repository for maintaining collaborative networks, cross-organizational collaborations, and their CBP models.	Public cloud only	Accessible by all organizations.
	IBP model repository	Repository for maintaining IBP models together with templates that organizations can define.	Public / private clouds	Accessible only by the owner organization.
	PAIS repository	Repository for maintaining the PAISs (the code of the process agents) of each organization and the executable specifications of the IBP models.	Public / private clouds	Accessible only by the owner organization.
Managers	Collaboration manager	It handles aspects related to the management of collaborative networks and CBP models. It also communicates with the IBP manager for generating the IBP models for each organization. It is the only component with access to the CBP model repository and also communicates with IBP managers for generating the IBP models for each organization.	Public cloud only	Accessible by all organizations.

	Component	Description	Location	Accessibility
	Collaboration manager stub	Private cloud side of collaboration manager in each private cloud, replicating the behavior of the latter for organizations relying on this cloud deployment model.	Private clouds only	Accessible only by the owner organization.
	IBP manager	It generates the IBP models for each organization from a CBP model and manages them, saving/retrieving these IBP models from the IBP repository.	Public / private clouds	Accessible only by the owner organization.
	PAIS manager	It manages the PAISs of each organization, generating, saving, or retrieving their implementation code and the corresponding executable IBP models from the PAIS repository. It also creates process agents when it is required, loading the executable IBP model into the process engine of the agents, and enacting these agents (and the IBPs, in consequence) during the execution of the CBP.	Public / private clouds	Accessible only by the owner organization.
Software agents	CBP monitoring agent	Software agent that monitors the CBP execution state. This state allows to determine if each organization involved in the collaboration has completed its part; who is waiting for some interaction; the global view of the CBP execution; etc.	Public cloud only	Accessible by all organizations.
	Process agent	Software agent that has an embedded process engine which executes the IBP model of each organization.	Public / private clouds	Accessible only by the owner organization.
	Agent to assist human tasks	Software agent that interacts with the organization to show up the proper forms to complete information by a human user.	Public / private clouds	Accessible only by the owner organization.
	Agent to assist automated tasks	Software agent that interacts with external programs or legacy systems, or that could require to invoke programs implementing decisions based on automated rules.	Public / private clouds	Accessible only by the owner organization.
General	Elasticity controller	It increases or reduces the resources of the infrastructure to be used by the whole cloud platform, according to the organization's needs.	Public / private clouds	
	Platform configurator	It allows the deployment and configuration of the platform components over a cloud infrastructure, deciding the required resources for them.	Public / private clouds	
	Message broker	Endpoint that allows the information exchange between components in different clouds.	Public / private clouds	

	Component	Description	Location	Accessibility
Web applications	Web application for CBP management	Web application that allows accessing to the services for the definition of collaborative networks and the implementation of CBPs –i.e. services offered by collaboration manager.	Public / private clouds	
	Web application for IBP management	Web application that interacts with IBP manager for accessing to the IBP model repository.	Public / private clouds	
	Web application for process execution	Web application that allows configuring the IBP model generated once the CBP model has been agreed, to become it the fully executable model – enacted by the process agent–. It also shows the forms to complete information by a human user when an IBP model is in execution by a process agent, so this application is linked to the agents to assist human tasks.	Public / private clouds	
	Web application for monitoring	Web application that shows the information provided by the CBP monitoring agents.	Public / private clouds	
	Web application for platform configuration	Web application that serves as the front-end to configure the platform for its deployment and specify aspects like the elasticity management.	Public / private clouds	

4.4 Interaction Between Components of the Platform

By using the proposed platform, a collaboration can be started at any moment by organizations that take part in a collaborative network. Figure 5 shows the interaction among the components for establishing a collaboration. In the left part of the figure are shown the components allocated in the *public cloud*; for instance, *orgA* and *orgB* are organizations using the public services. The right part of the figure shows the components allocated in a *private cloud*, owned by the organization *orgC*.

The organization that initiates the collaboration (orgA) starts a simple negotiation process sending to the collaboration manager a *newCollaboration* message, indicating the collaborative network, and proposing a collaboration agreement and a CBP model. Next, each organization in the collaboration (except orgA) receives a *notify* message from the collaboration manager with a reference of the collaboration that might participate in. For the organizations that are making use of a private cloud, the collaboration manager forwards this message to the corresponding stub on private clouds (via the message broker). This negotiation process can derive in different scenarios, since organizations could answer by accepting or rejecting to collaborate. In an ideal situation, all organizations agree to collaborate sending an *agree* message to the collaboration manager. Organizations with private clouds do the proper to the stubs; then, each stub retrieves these agreement messages to the collaboration manager in the public cloud. As it can be noticed, methods of the collaboration manager stub are stereotyped by «*echo*» and «*gathering*». The «echo» stereotype means a message "replication" from collaboration manager in the public cloud and the «gathering» stereotype means that information is gathered to the collaboration manager.

Next, for each organization involved in the collaboration in public cloud, the collaboration manager sends a *generateIBPModel* message to the IBP manager, which creates an incomplete IBP model for each organization. To do that, the IBP manager employs the method and tool proposed in [23]. These incomplete models are saved into the IBP model repository of each organization via the *saveIBPModel* method. Collaboration manager also sends a generateIBPModel message to each private cloud, so collaboration manager stub can replicate this process with local IBP managers and IBP model repositories.

For allowing organizations in both clouds to configure their incomplete IBP models with all the information and resource links necessary to become fully executable IBP models, the platform offers the services of the PAIS manager to perform this task, by invoking the *configureIBP* method (Figure 6). Then, the PAIS manager gets the IBP model from the IBP model repository; if the model exists, the organization is able to apply the modifications necessary to complete the model via the Web application for process execution. Once available the information to complete the IBP

model, an executable IBP model is generated invoking the *generateExecutableIBPModel* method. This model can be directly interpreted by a process agent and is saved into the PAIS repository using the *saveExecutableIBPModel* method.

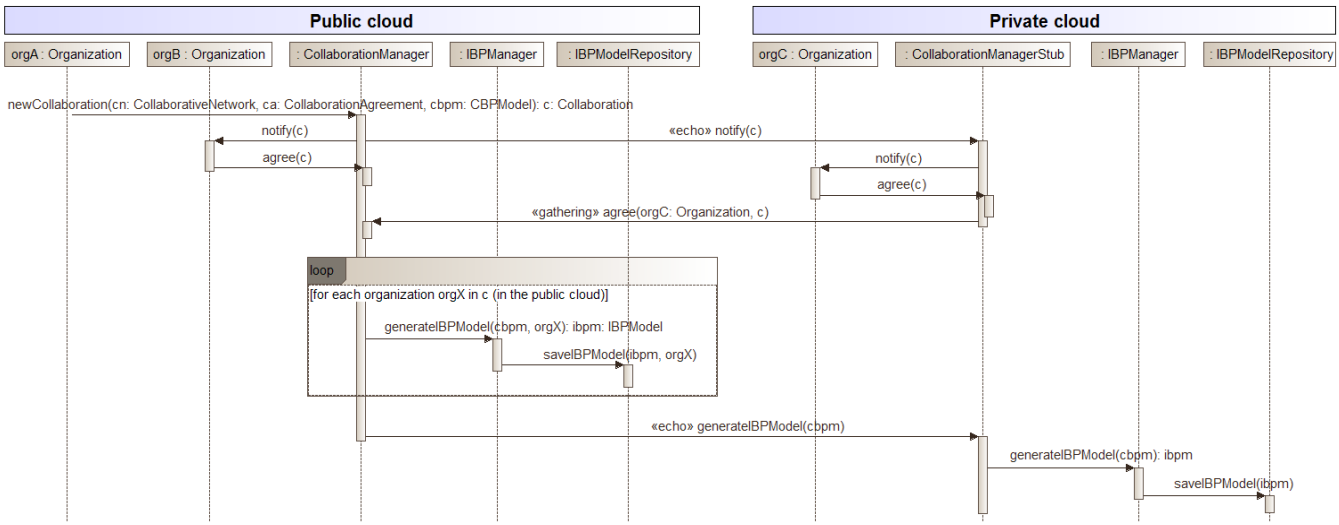


Figure 5: Beginning of a collaboration and generation of the initial IBP models.

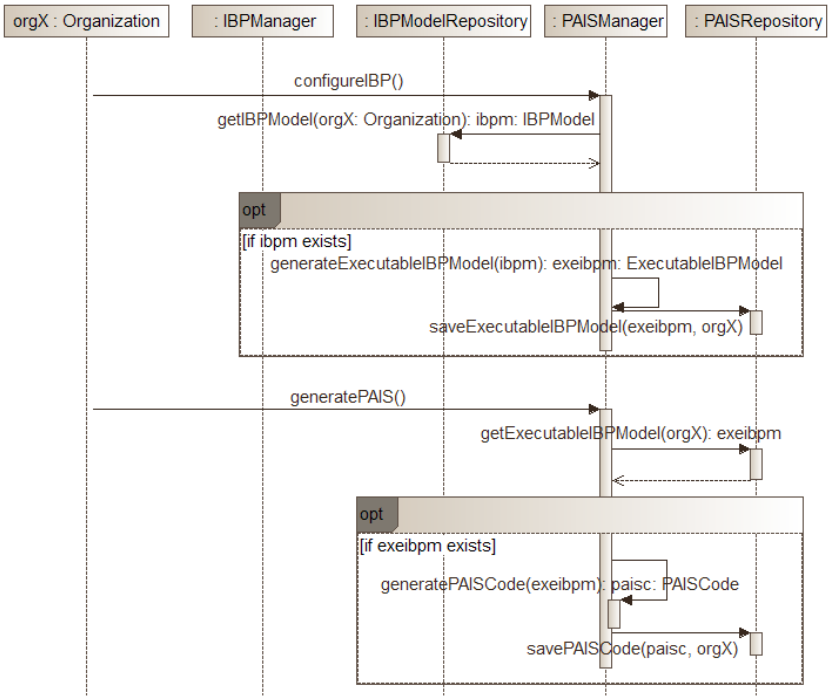


Figure 6: Configuration of IBP models and generation of the PAISs code.

From now on, each organization is able to generate the PAIS code for the process agent that will have embedded the process engine for enacting the executable IBP model. This process starts by invoking the *generatePAIS* method. Then, the PAIS manager gets the executable IBP model via the PAIS repository and, in case it exists, generates the PAIS code by invoking the *generatePAISCode* method; this is done reusing methods from [8]. The code is saved into the PAIS repository by invoking the *savePAISCode* method. The code of the executable IBP model corresponds to BPMN interchange file (in XML format) defined with an executable BPMN version provided by the Jadex Processes framework. The generated code of the agent is defined in an Agent Definition File (ADF) based on a XML Format, which follows the structure of the Jadex metamodel. An ADF contains the roles, beliefs, goals, plans, events, and configuration parameters of an agent to enable its enactment in the Jadex platform. In the configuration of the roles of the agent, the behavior of a role can be specified by an executable BPMN model. Hence, the PAIS code generated by the platform involves the ADF file of the agent and the executable BPMN file that implement the behavior of the role of the agent in the CBP.

Once the code of the PAIS is available, each organization is able to enact their IBPs. Again, this is done in the same way either in the public or private cloud. The sequence of interactions between the components is illustrated in Figure 7. Organizations can make use of the PAIS manager to enact IBPs by invoking the *enactIBP* method. The PAIS manager first gets the PAIS code and executable IBP model, retrieving both things from the PAIS repository (method *getPAISCode*) and the IBP model repository (method *getExecutableIBPModel*), respectively. Next, if both of them exist, a process agent (*procAgX*) is created by the *createProcessAgent* method. After that, the process agent executes a sequence of tasks: it loads the executable IBP model; enacts the model; and finally, creates a software agent to monitor the execution of the CBP (*cbpMonAg*), if this agent was not previously created by the enactment of another IBP).

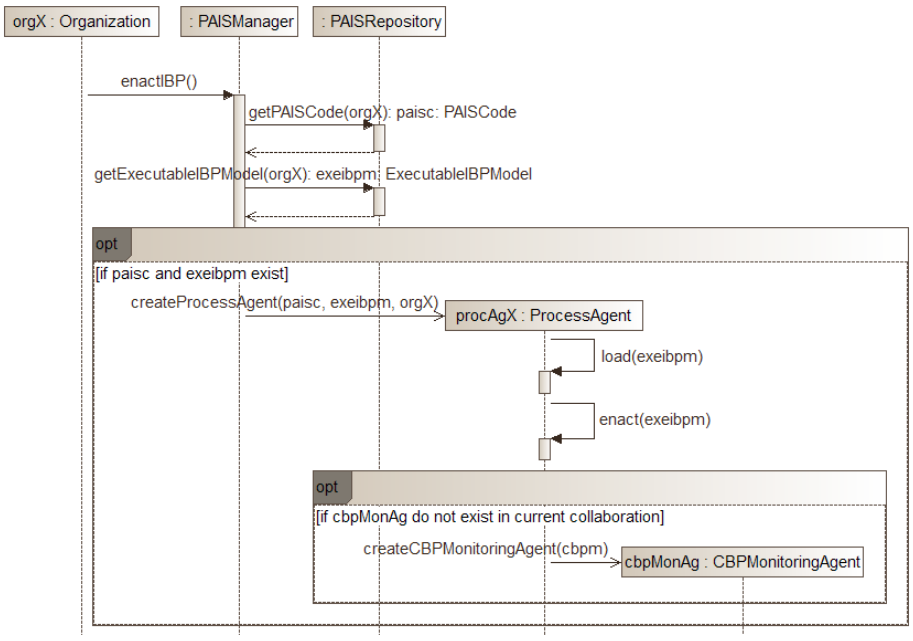


Figure 7: Enactment of IBPs for CBP execution.

During the execution of the CBP, each process agent representing an organization is able to interact with other process agents –which represent the rest of the organizations–, in order to exchange the information necessary to carry out the collaboration. To implement the private tasks of the IBP, auxiliary software agents are created: agents to assist human tasks and agents to assist automated tasks, as it was described in subsection 4.2.

All public information referent to IBPs is updated by the CBP monitoring agent (cbpMonAg) so any of the organizations can follow the general state of the collaboration and check the information reported by the other organizations.

Each process agent in the collaboration expires once completed the execution of its IBP. However, the PAIS code is available if it is required to create a new process agent later because of a new instance of an IBP has to be managed. The CBP monitoring agent (cbpMonAg) also expires when the collaboration and CBP finish.

4.5 Implementation Schema of the Platform

The platform is intended to be deployed by using an IaaS infrastructure, so one important aspect is the deployment and management system to be used to orchestrate the platform components and allow a PaaS solution.

In order to achieve the requirement of portability, it is proposed to implement the platform by following a standard for deployment and management of PaaS platform's lifecycles. Even though initially was considered the *OCCI-PaaS* standard [43] in previous work [11], the lack of implementations besides the *rOCCI* project [44], lead us to the selection of another alternative (there is a proposal of a Java implementation called *jOCCI*, but uses the services of the rOCCI implementation behind the scenes [45]). The *Cloud Application Management for Platforms* (*CAMP*) [46] standard was selected to implement the platform, since it is oriented to PaaS solutions and focused not only the deployment but also in the lifecycle management of the deployed applications. As an implementation of the CAMP standard, *Apache Brooklyn*[1] was selected. Apache Brooklyn is an opensource project powered by Apache that offers an environment for deploying cloud applications over a wide selection of cloud providers or cloud platforms for IaaS level. This interaction is possible through the *Apache jclouds*[2] library, another opensource project from Apache based on Java that offers an abstraction and homogenization of APIs from several cloud providers and platforms. Apache Brooklyn provides a framework for defining cloud applications, selecting the cloud providers or platforms for the deployment, and also offers a monitoring environment supporting the management of the lifecycles of the applications that had been deployed, allowing a run-time management according to the CAMP standard [47].

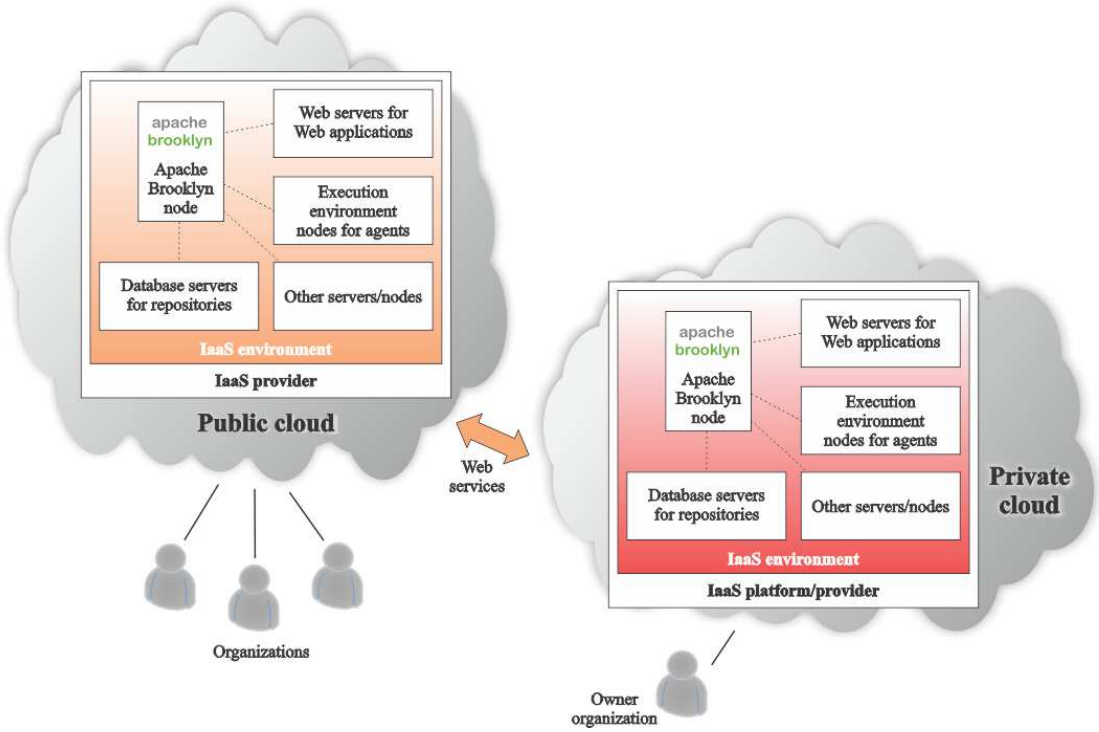


Figure 8: Implementation scheme of the cloud-based platform for executing CBPs.

[1] https://brooklyn.apache.org/

[2] http://jclouds.apache.org/

Hence, the proposed implementation schema is based on the use of Apache Brooklyn and the CAMP standard to make the deployment of the platform on a IaaS environment. This allows the deployment could be done in different IaaS providers, such as Amazon, Google and OpenStack. In addition, Apache Brooklyn interacts with the IaaS environment to create the required virtual machines according to the deployment plan or model, so it plays the role of the *platform configurator* component proposed in our architecture. It also supports the management of the application's lifecycles, acting as the *elasticity controller* and the *Web application for platform configuration*. Apache Brooklyn can be installed as another virtual machine (node) on the IaaS environment, as can be seen in Figure 8.

Figure 8 shows the implementation schema with the nodes that are required to make the deployment of the platform. The nodes refer to the virtual machines to be created and the components of the platform, along with the required applications for them, that have to be allocated in these virtual machines. CBP model repository, IBP model repositories, and PAIS repositories components are allocated in virtual machines that contain database servers, in order to manage these repositories in databases. All Web applications are allocated in virtual machines containing Web servers. Software agents are deployed on virtual machines containing the Jadex agent execution environment.

The Apache Brooklyn node allows monitoring and controlling in real time if the resources (the created virtual machines) are enough for the components that are being executed, and it creates new virtual machines (or removes them) according to the elasticity strategy it provides.

5 Case Study

The aim of this section is to describe the functionality and applicability of the proposed cloud-based platform for executing CBPs with an implementation of a study case from a distribution network of electronic products, which was presented in [7]. This distribution network is a collaborative network consisting of: *Megatronic*, a retailer with points of sales around the center and east regions of Argentina; and *Philkaw*, *Grundrive*, and *Sanx*, which are assemblers of electronic products and suppliers of the retailer. Suppliers collaborate with the retailer in a separate and peer-to-peer way. Each supplier established an independent cross-organizational collaboration with the retailer to carry out a particular CBP model. Megatronic agreed to carry out a CPFR model with Philkaw and Grundrive, and a VMI model with Sanx. VMI (*Vendor Managed Inventory*) [48] is another collaborative business model to build demand-driven supply chains, as CPFR.

The deployment of the platform is as follow: Megatronic and Philkaw use a public cloud, and Grundrive and Sanx implement private clouds. The public cloud is deployed over an *Amazon EC2*[3] infrastructure (IaaS) –even though any other cloud provider like *Google Compute Engine*[4] can be chosen. Amazon EC2 represents a virtual computing environment, allowing to use Web service interfaces to launch virtual machine instances with a variety of operating systems, load them with custom application environment, and manage the virtual networks where the virtual machines will be connected.

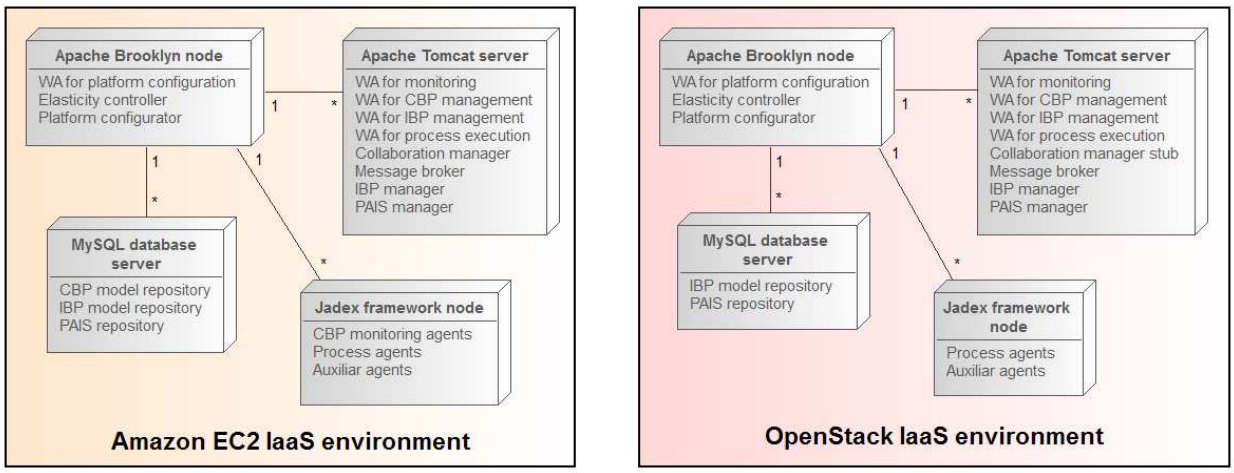
For the private clouds, Grundrive and Sanx use an *OpenStack*[5] infrastructure. Any other IaaS platform could be also employed as well, e.g. *Apache CloudStack*[6], but for simplicity purposes, it will be the same for both organizations. The OpenStack project is an open source cloud computing platform that provides an IaaS solution like Amazon EC2 through a wide variety of complementary services. Hence, Amazon EC2 and OpenStack are the IaaS environments used and where the virtual machines are deployed and executed, to give support to the whole architecture of the cloud-based platform for CBPs. An *Apache Brooklyn node* is configured and executed as another virtual machine instance on each cloud, which acts as the Web application for platform configuration, the platform configurator, and elasticity controller components defined in the platform architecture. The rest of the nodes are shown in Figure 9, where there are deployments schemas for each cloud.

[3] https://aws.amazon.com/ec2/

[4] https://cloud.google.com/compute

[5] https://www.openstack.org/

[6] https://cloudstack.apache.org/



(a) Public cloud deployment. (b) Private clouds deployment.

Figure 9: Deployment of the platform for CBP management according to the situation described in the case study.

About specific technologies used to implement the remaining components of the architecture, the Web applications are developed using *JavaServer Pages* (*JSP*) to be supported by *Apache Tomcat servers*. Elasticity for these applications can follow a strategy based on Web requests per second. When a limit of requests per second is achieved, a new instance of the server is created, else if the requests decrease that threshold, unnecessary instances are terminated. As can be seen in Figure 9, the collaboration manager and the message broker in public cloud are included as services in the Web application for CBP management. In the same way, the collaboration manager stubs and the message brokers in private clouds are included as services in the Web application for CBP management deployed in the two private clouds (these Web applications are different from the one deployed in public cloud). IBP managers and PAIS managers are included as part of the Web application for IBP management and Web application for process execution, respectively.

Databases for all repositories are implemented using *MySQL database servers*, which are compatible with repositories implemented in [7], as well as the Web applications. The process agents are implemented as software agents by using the Jadex framework and platform, and the embedded process engine of the agents used to enact the IBPs of each organizations is implemented via the Jadex Processes framework, as was detailed before in subsection 4.2. *Jadex framework nodes* are used to this purpose, using its own elasticity strategy; the rest of the software agents also use these nodes.

As an example of a deployment configuration of the public cloud of the platform, Figure 10 shows part of an Apache Brooklyn script (known as *blueprint*), specified in *YAML*[7], for deploying and configuring the nodes (virtual machines) for the Apache Tomcat server in public cloud. Lines 9 to 37 describe the services to be deployed. Line 39 ("`location`") specifies the IaaS provider or platform that will be used –Amazon EC2 in this case. Line 10 ("`serviceType`") specifies the global service type, a cluster of nodes for Web applications. From line 13 to 27 there are several configuration parameters for deployment. "`initialSize`" (line 14) indicates the initial size of the cluster (one node in the example). Next, lines 15 to 19 allow to specify the binaries of the Web applications that implement the components which offer the services (Web application for monitoring, Web application for CBP management, Web application for IBP management, and Web application for process execution, respectively). Lines 21 to 23 detail the execution environment for the applications, which will be installed on each node (Apache Tomcat v8 server). Lines 25 to 27 allow setting a load balancer (*nginx*[8]). From line 29 there are several policies, related to elasticity in this case. Here, it is important to highlight lines 32 to 37. Line 32 selects the metric of the elasticity rule: *Web requests per second*. Lines 33 and 34 are the upper and lower threshold, respectively. Stabilization delays for triggering are set in

[7] http://yaml.org/

[8] https://nginx.org/en/

lines 35 and 36. Finally, line 37 ("`maxPoolSize`"), limits the maximum number of instances of the cluster to a fixed value (5 instances).

```
 1 brooklyn.catalog:
 2   version: 0.0.1
 3   items:
 4   - id: ibp-management-nodes
 5     description: IBP management nodes
 6     itemType: template
 7     item:
 8       name: CBP management nodes
 9       services:
10       - serviceType: org.apache.brooklyn.entity.webapp.ControlledDynamicWebAppCluster
11         id: cluster-ibp-management-nodes-tomcat-v8
12         name: "Cluster of IBP management nodes running Apache Tomcat servers v8"
13         brooklyn.config:
14           initialSize: 1
15           wars:
16             - /home/ubuntu/apache-tomcat-8.5.14/wa-mon.war
17             - /home/ubuntu/apache-tomcat-8.5.14/wa-cbp-mng.war
18             - /home/ubuntu/apache-tomcat-8.5.14/wa-ibp-mng.war
19             - /home/ubuntu/apache-tomcat-8.5.14/wa-proc-exec.war
20           ...
21           memberSpec:
22             $brooklyn:entitySpec:
23               type: org.apache.brooklyn.entity.webapp.tomcat.Tomcat8Server
24           ...
25           controlleddynamicwebappcluster.controllerSpec:
26             $brooklyn:entitySpec:
27               type: org.apache.brooklyn.entity.proxy.nginx.NginxController
28               ...
29         brooklyn.policies:
30         - type: org.apache.brooklyn.policy.autoscaling.AutoScalerPolicy
31           brooklyn.config:
32             metric: webapp.reqs.perSec.perNode
33             metricUpperBound: 10
34             metricLowerBound: 1
35             resizeUpStabilizationDelay: 2s
36             resizeDownStabilizationDelay: 1m
37             maxPoolSize: 5
38       ...
39       location: amazon-ec2
```

Figure 10: YAML blueprint for deploying and configuring the Apache Tomcat server via Apache Brooklyn, making use of the Amazon EC2 cloud service (public cloud).

Using the Cloud-based platform for executing CBPs, first, the retailer accesses to the Web application for CBP management of the public cloud and creates the collaborative network *Electronic Products Collaborative Distribution*. Then the retailer looks for suppliers in the platform and sent them an invitation to join this network. Afterwards, the suppliers join in the collaborative network created by the retailer (see in Figure 11 the *Collaborative Network Members* shown in screenshot of the Web application for CBP management). Then, the retailer creates three cross-organizational collaborations which refer to collaborations defined by the retailer with each supplier (see in Figure 11 VMI-based Collaboration with Philkaw, VMI-based Collaboration with Grundrive, and CPFR-based Collaboration with Sanx). Each collaboration indicates the business model adopted and the role the organizations fulfill.

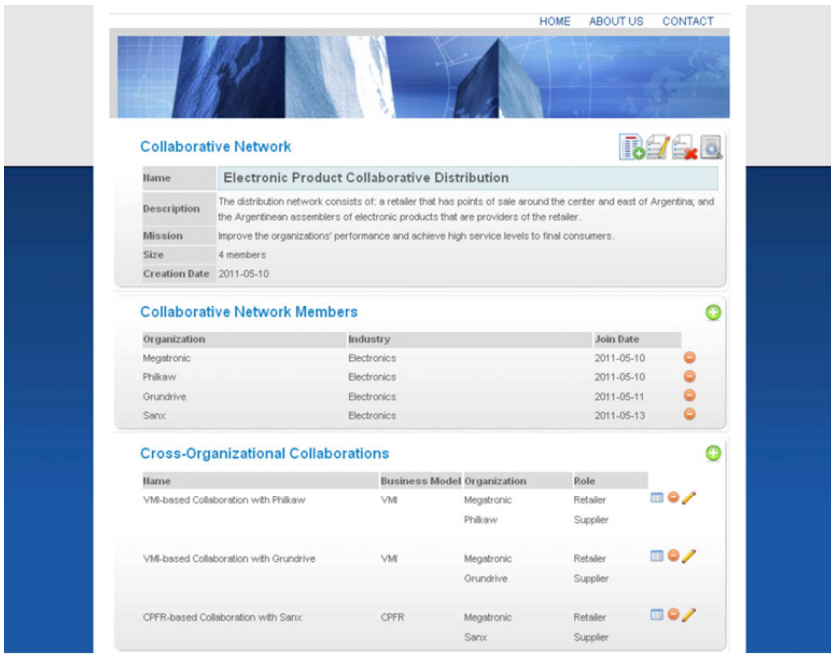


Figure 11: Web application for CBP management, application reused from [7].

For each one of the created collaborations, organizations can manage the CBP models of the CBPs to be executed by using the Web application for CBP management, which interacts with the collaboration manager to store the CBP models into the CBP repository. As an example, the retailer added several CBP models to the CBP model repository as part of the "CPFR-based Collaboration with Sanx" cross-organizational collaboration, for instance the Collaborative Replenishment Plan Management model (CPFR process). Figure 12 shows the interaction protocol specified with the UP-ColBPIP language, which defines the behavior of this CBP model. This protocol manages a simple negotiation process between the retailer and the supplier for agreeing on a replenishment plan of several products for a short-time period. The protocol starts with the supplier that proposes a supply plan to the retailer, who evaluates it and decides to reject, accept or make a counter-proposal. The decision is sent to the supplier. In case of rejection or acceptance, the process finishes. In case of a counter-proposal, the supplier evaluates it and responds to the retailer with an acceptance or rejection.

By adding this CBP model to the cross-organizational collaboration with the Web application for CBP management, then Collaboration Manager communicates with the IBP Manager of the organization context of the organizations involved in that CBP, i.e. with the IBP Manager of the "Megatronic" retailer's organization context and the IBP Manager if the "Sanx" supplier's organization context. These IBP Manager are in charge of generating the IBP model of each organization involved in the CBP. The generated IBP models are stored in each one of the IBP repositories (via methods of the IBP manager). Figure 13 shows a BPMN model representing the generated IBP model for the retailer.

From the generated IBP models and by means of the Web application for process execution, organizations configure their corresponding executable IBP models and the code for the process agents that will be responsible for executing them, and add them to their PAIS repositories.

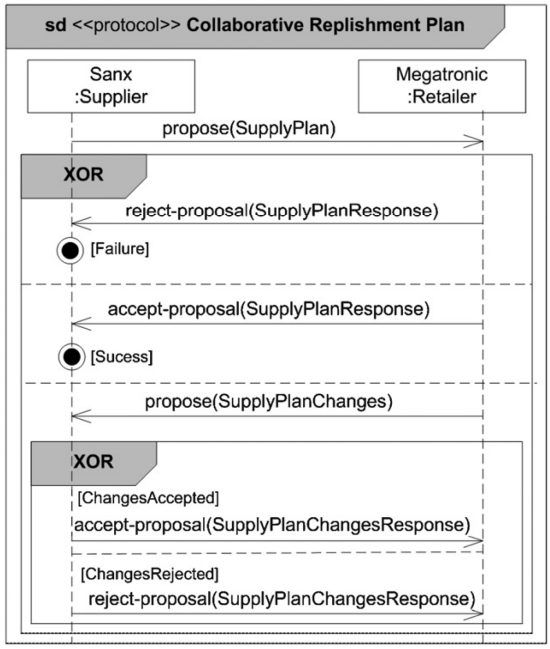


Figure 12: Collaborative Replenishment Plan CBP model (from [7]).

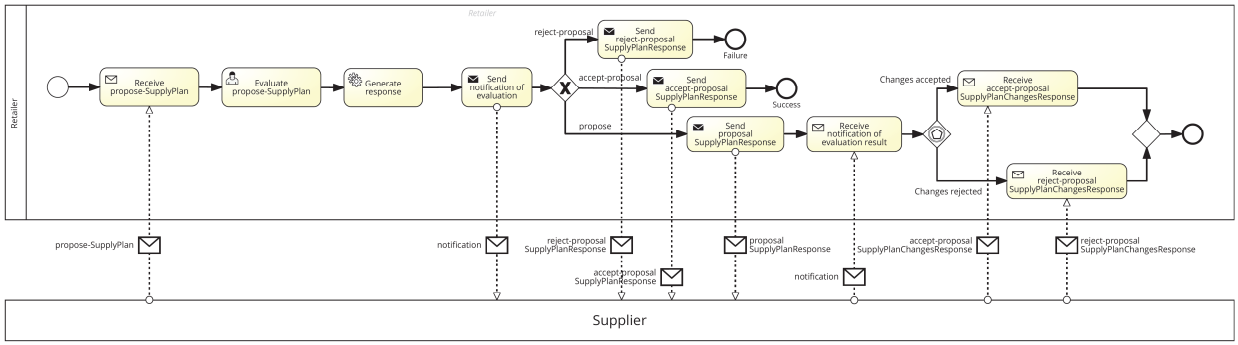


Figure 13: Retailer's IBP model (from [7]).

Web application for process execution allows organizations to execute the IBP processes too. The execution implies accessing the PAIS repository (via the PAIS manager), retrieving the executable IBP model and the PAIS code for the agent, creating the process agent instance and all the auxiliar agents' instances needed for the process execution, and creating the CBP monitoring agent instance, if it wasn't created before by the execution of another IBP in the same CBP.

Once in execution the IBPs (and the CBP, in consequence), each organization can follow the state of the collaboration using the Web application for monitoring. This is particularly useful for the retailer, which can be aware of the progress of the collaboration established with each provider.

6 Conclusions and Future Works

This work proposed a cloud-based platform to execute collaborative business process (CBPs) with the aim to support dynamic and agile collaborative networks. In these networks is not only required the on-demand execution of CBPs but also it is required the on-demand generation and deployment of new PAISs, to implement and execute new CBPs the organizations want to carry out as part of their collaborative networks. The proposed platform fulfills this by providing cloud services for the initiation of collaborations and execution of CBPs, as well as cloud services that allow generating and deploying the PAISs of the organizations after the addition of CBP model in the platform. Thus, the platform enables the implementation, execution, and monitoring of CBPs in agile collaborative networks. In addition, the use of cloud computing to execute CBPs enable the organizations can leverage collaborative networks in minor times and costs since they do not require to create their own IT infrastructures for deploying the PAISs required to execute CBPs.

The defined platform architecture allows fulfilling the main functional requirements for CBP execution: organization autonomy, decentralized execution, global view of message exchange, and peer-to-peer interactions.

Organization autonomy is achieved in two ways by the platform: (1) keeping services and repositories owned by one organization in a context that preserves its privacy, which guarantees that the organization can manage services for generating its models (IBPs) and execute them without any centralized approach, in an autonomous way; (2) and by using software agents that implement the PAISs of the organizations and enable them to execute their IBPs in an autonomous way.

The decentralized execution of CBPs is supported by using software agents that implement the PAISs of the organizations and enable them to execute their IBPs in an autonomous way, interacting with the PAISs of the other organizations to carry out a distributed and decentralized execution of CBPs. Also, some centralization only is required for managing CBP repositories, because CBP models need to be known, shared and available for all organizations.

Peer-to-peer interactions are also achieved with the platform, which provides a message broker that enable the message interchange between the organizations, so monitoring and process agents can share public information between them. By means of the cloud infrastructure, it is also possible facilitate the global view of message exchange: all messages travel through the cloud (by the message brokers) and can be caught or received by the interested components.

Cloud computing technology not only gives support to these requirements too, but also meets an additional advantage: possibility of offering the services on-demand, paying only for their use. The use of cloud computing allows offering services on-demand to execute CBPs, paying only for their use for the organizations. This enables the organizations can leverage collaborative networks in minor times and costs since they do not require to create their own IT infrastructures for deploying the PAISs required to execute CBPs. However, cloud computing solutions brings with additional issues about non-functional requirements that have direct influence on the solutions. In this work three non-functional requirements were identified as important and handled: (1) elasticity, (2) privacy, and (3) portability issues.

The elasticity degree of the cloud services for CBP execution is an important feature to make a distinction of cloud platforms from other similar approaches. The proposed platform deals with it by controlling elasticity at the level of the components that execute the processes –i.e. at the level of the agent-based PAISs. This allows flexibility for providing good performance, considering the necessary resources for the execution of each process agent and IBP instance.

Privacy considerations have been taken into account during the design of the platform. This is the main reason why the platform offers a deployment model on a public cloud and private clouds. Organizations can maintain control over all their sensitive information, internal processes, etc., and they only have to provide public information used to determine the global state of the collaboration. Besides, appropriate organization contexts provide isolation for each organization that uses the public service (shared among several organizations), allowing repository spaces and execution environments for process agents accessible only by the owner organization.

Portability is achieved through the platform configurator component and the elasticity controller, which are specified by making use of the CAMP standard, the jclouds library, and the Apache Brooklyn implementation. This enables the cloud-based platform for executing CBPs is portable and can be deployed in different IaaS clouds provided by different vendors. Thus, a third-party that wants to provide the platform can make use of any of the cloud providers/platforms supported by the jclouds library. For private clouds, organizations can implement the CBP platform by using cloud platforms also supported by jclouds. This allows organizations or third-party providers to select the underlying cloud infrastructure and decide it in terms of costs, availability, or other features, without dealing with the problems of technologies used.

Future work is concerned about the definition of elasticity mechanisms to offer efficient on-demand services for the CBP execution. Also it is expected to include in the platform methods and tools to support to the stages of evaluation and analysis of CBP processes, such as process mining and simulation of CBPs, to improve them. Finally, we will focus on conclude and validate the platform through the implementation of real cases from different domains such as *e-healthcare* or *e-government*.

References

[1] Chituc, C. M., Azevedo, A., & Toscano, C. (2009). "A framework proposal for seamless interoperability in a collaborative networked environment". *Computers in industry*, 60(5), 317-338.

[2] Camarinha-Matos, L. M., Afsarmanesh, H., Galeano, N., & Molina, A. (2009). "Collaborative networked organizations–Concepts and practice in manufacturing enterprises". *Computers & Industrial Engineering*, 57(1), 46-60.

[3] Andres, B., Macedo, P., Camarinha-Matos, L. M., & Poler, R. (2014, October). "Achieving coherence between strategies and value systems in collaborative networks". *In Working Conference on Virtual Enterprises* (pp. 261-272). Springer Berlin Heidelberg.

[4] Villarreal, P. D., Salomone, E., & Chiotti, O. (2007). "Modeling and Specification of Collaborative Business Processes with a MDS Approach and a UML Profile". *In Enterprise modeling and computing with UML* (pp. 13-44). IGI Global.

[5] Weske, M. (2012). "*Business process management: concepts, languages, architectures*" (2nd. Edition). Springer Publishing Company, Incorporated.

[6] Object Management Group, OMG. (2011). "*Business Process Model and Notation version 2.0*". Specification "formal/2011-01-03". Object Management Group. http://www.omg.org/spec/BPMN/2.0/PDF/.

[7] Lazarte, I. M., Thom, L. H., Iochpe, C., Chiotti, O., & Villarreal, P. D. (2013). "A distributed repository for managing business process models in cross-organizational collaborations". *Computers in Industry*, 64(3), 252-267.

[8] Tello-Leal, E., Chiotti, O., & Villarreal, P. D. (2014). "Software agent architecture for managing inter-organizational collaborations". *Journal of applied research and technology*, 12(3), 514-526.

[9] Gupta, P., Seetharaman, A., & Raj, J. R. (2013). "The usage and adoption of cloud computing by small and medium businesses". *International Journal of Information Management*, 33(5), 861-874.

[10] Grefen, P. (2013). "Networked business process management". *International Journal of IT/Business Alignment and Governance (IJITBAG)*, 4(2), 54-82.

[11] Cocconi, D., Roa, J., & Villarreal, P. D. (2017). "Cloud-based platform for collaborative business process management". *In Simposio Latinoamericano de Procesos de Negocio, Arquitecturas y Sistemas Organizacionales (SLPNASO)-JAIIO 46* (Córdoba, 2017).

[12] Pallis, G. (2010). "Cloud computing: the new frontier of internet computing". *IEEE internet computing*, 14(5), 70-73.

[13] Lin, A., & Chen, N. C. (2012). "Cloud computing as an innovation: Perception, attitude, and adoption". *International Journal of Information Management*, 32(6), 533-540.

[14] Dillon, T., Wu, C., & Chang, E. (2010, April). "Cloud computing: issues and challenges". *In Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on* (pp. 27-33). IEEE.

[15] Han, R., Ghanem, M. M., Guo, L., Guo, Y., & Osmond, M. (2014). "Enabling cost-aware and adaptive elasticity of multi-tier cloud applications". *Future Generation Computer Systems*, 32, 82-98.

[16] Calero, J. M. A., Edwards, N., Kirschnick, J., Wilcock, L., & Wray, M. (2010). "Toward a multi-tenancy authorization system for cloud services". *IEEE Security & Privacy*, 8(6), 48-55.

[17] Azeez, A., Perera, S., Gamage, D., Linton, R., Siriwardana, P., Leelaratne, D., ... & Fremantle, P. (2010, July). "Multi-tenant SOA middleware for cloud computing". *In Cloud computing (cloud), 2010 IEEE 3rd international conference on* (pp. 458-465). IEEE.

[18] Tang, J., Cui, Y., Li, Q., Ren, K., Liu, J., & Buyya, R. (2016). "Ensuring security and privacy preservation for cloud data services". *ACM Computing Surveys* (CSUR), 49(1), 13.

[19] Silva, G. C., Rose, L. M., & Calinescu, R. (2013, December). "Towards a model-driven solution to the vendor lock-in problem in cloud computing". *In Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on* (Vol. 1, pp. 711-716). IEEE.

[20] Petcu, D., & Vasilakos, A. V. (2014). "Portability in clouds: approaches and research opportunities". *Scalable Computing: Practice and Experience*, 15(3), 251-270.

[21] Villarreal, P. D., Salomone, E., & Chiotti, O. (2006). "A MDA-based development process for collaborative business processes". *Milestones, Models and Mappings for Model-Driven Architecture*, 17.

[22] Villarreal, P. D., Lazarte, I. M., Roa, J., & Chiotti, O. (2009). "A Modeling Approach for Collaborative Business Processes Based on the UP-ColBPIP Language". *In Business Process Management Workshops* (Vol. 43, pp. 318-329).

[23] Lazarte, I. M., Tello-Leal, E., Roa, J., Chiotti, O., & Villarreal, P. D. (2010, October). "Model-driven development methodology for B2B collaborations". *In Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2010 14th IEEE International* (pp. 69-78). IEEE.

[24] del–Río–Ortega, A., Gutiérrez, A. M., Durán, A., Resinas, M., & Ruiz–Cortés, A. (2015, June). "Modelling service level agreements for business process outsourcing services". *In International Conference on Advanced Information Systems Engineering* (pp. 485-500). Springer, Cham.

[25] Yu, D., Zhu, Q., Guo, D., Huang, B., & Su, J. (2015, June). "jBPM4S: A multi-tenant extension of jBPM to support BPaaS". *In Asia-Pacific Conference on Business Process Management* (pp. 43-56). Springer International Publishing.

[26] Mohamed, M., Amziani, M., Belaïd, D., Tata, S., & Melliti, T. (2015). "An autonomic approach to manage elasticity of business processes in the Cloud". *Future Generation Computer Systems*, 50, 49-61.

[27] Han, Y. B., Sun, J. Y., Wang, G. L., & Li, H. F. (2010). "A cloud-based bpm architecture with user-end distribution of non-compute-intensive activities and sensitive data". *Journal of Computer Science and Technology*, 25(6), 1157-1167.

[28] Schulte, S., Janiesch, C., Venugopal, S., Weber, I., & Hoenisch, P. (2015). "Elastic business process management: state of the art and open challenges for BPM in the cloud". *Future Generation Computer Systems*, 46, 36-50.

[29] Woitsch, R., & Utz, W. (2015, October). "Business process as a service: Model based business and IT cloud alignment as a cloud offering". *In Enterprise Systems (ES), 2015 International Conference on* (pp. 121-130). IEEE.

[30] Pokahr, A., & Braubach, L. (2015). "Elastic component-based applications in PaaS clouds". *Concurrency and Computation: Practice and Experience*.

[31] Bentounsi, M., Benbernou, S., & Atallah, M. J. (2012, June). "Privacy-preserving business process outsourcing". *In Web Services (ICWS), 2012 IEEE 19th International Conference on* (pp. 662-663). IEEE.

[32] Accorsi, R. (2011, July). "Business process as a service: Chances for remote auditing". *In Computer Software and Applications Conference Workshops (COMPSACW)*, 2011 IEEE 35th Annual (pp. 398-403). IEEE.

[33] Camarinha-Matos, L. M., Juan-Verdejo, A., Alexakis, S., Bär, H., & Surajbali, B. (2015, February). "Cloud-based

collaboration spaces for enterprise networks". *In Computing and Communications Technologies (ICCCT), 2015 International Conference on* (pp. 185-190). IEEE.

[34] Benaben, F., Mu, W., Boissel-Dallier, N., Barthe-Delanoe, A. M., Zribi, S., & Pingaud, H. (2015). "Supporting interoperability of collaborative networks through engineering of a service-based Mediation Information System (MISE 2.0)". *Enterprise Information Systems*, 9(5-6), 556-582.

[35] Sun, Y., Su, J., & Yang, J. (2014, September). "Separating execution and data management: A key to business-process-as-a-service (BPaaS)". *In International Conference on Business Process Management* (pp. 374-382). Springer, Cham.

[36] Küster, T., Lützenberger, M., Heßler, A., & Hirsch, B. (2012). "Integrating process modelling into multi-agent system engineering". *Multiagent and Grid Systems*, 8(1), 105-124.

[37] Min, H., & Yu, W. B. V. (2008). "Collaborative planning, forecasting and replenishment: demand planning in supply chain management". *International Journal of Information Technology and Management*, 7(1), 4-20.

[38] Alt, R., Gizanis, D., & Legner, C. (2005). "Collaborative order management: toward standard solutions for interorganisational order management". *International Journal of Technology Management*, 31(1-2), 78-97.

[39] Roa, J., Villarreal, P. D., & Chiotti, O. (2012). "Behavior Alignment and Control Flow Verification of Process and Service Choreographies". *J. UCS*, 18(17), 2383-2406.

[40] Roa, J., Reynares, E., Caliusco, M.L., Villarreal, P. (2017). "Ontology-based Heuristics for Process Behavior: Formalizing False Positive Scenarios". *Lecture Notes in Business Information Processing*. Vol. 281. To appear.

[41] Comuzzi, M., & Angelov, S. (2016). "Patterns and tools for business process monitoring customization". *Service Oriented Computing and Applications*, 10(3), 253-271.

[42] Pokahr A., Braubach L., Lamersdorf W. (2005) "Jadex: A BDI Reasoning Engine". *In: Bordini R.H., Dastani M., Dix J., El Fallah Seghrouchni A. (eds) Multi-Agent Programming. Multiagent Systems, Artificial Societies, and Simulated Organizations* (International Book Series), vol 15. Springer, Boston, MA.

[43] Metsch, T., & Mohamed, M. (2016). "*Open Cloud Computing Interface - Platform*". Specification "GFD.227". Open Grid Forum. http://ogf.org/documents/GFD.227.pdf.

[44] García, Á. L., del Castillo, E. F., & Fernández, P. O. (2016). "ooi: Openstack occi interface". *SoftwareX*, 5, 6-11. Elsevier.

[45] Kimle, M., Parák, B., & Šustr, Z. (2015, March). "jOCCI–General-Purpose OCCI Client Library in Java". *In International Symposium on Grids and Clouds (ISGC)* (Vol. 15, No. 20).

[46] "*Cloud Application Management for Platforms*". Version 1.1. Committee Specification 01. docs.oasis-open.org/camp/camp-spec/v1.1/camp-spec-v1.1.html.

[47] Carrasco, J., Cubo, J., Durán, F., & Pimentel, E. (2016, June). "Bidimensional cross-cloud management with TOSCA and Brooklyn". *In Cloud Computing (CLOUD), 2016 IEEE 9th International Conference on* (pp. 951-955). IEEE.

[48] Franke, P. D. (2010). "*Vendor-managed inventory for high value parts: results from a survey among leading international manufacturing firms*" (Vol. 3). Univerlagtuberlin.