Cloud-based Platform for Collaborative Business Process Management

Diego Cocconi, Jorge Roa, and Pablo Villarreal

Abstract—With the wide adoption of the Internet, organizations establish collaborative networks to execute Collaborative Business Processes (CBPs). Current approaches of Process-Aware Information Systems (PAISs) to implement and execute CBPs have shortcomings: high costs and complexity of IT infrastructure to deploy the PAISs; poor support for autonomy, decentralization, global view of message exchange and peer-to-peer interactions; and rigid platforms for generating and deploying PAISs on-demand according with the CBPs agreed in collaborative networks. To overcome these issues, this work proposes a cloud-based platform for the management of CBPs. The platform provides cloud services that enable the generation and deployment on-demand of the PAISs required to implement the agreed CBPs, as well as the execution on-demand of CBPs by fulfilling the abovementioned issues. To deal with privacy issues, the platform can be deployed in private clouds. Elasticity is provided at the level of process instances and portability is also achieved.

Index Terms—cloud computing, business process, inter-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 of organizations is established and carried out through *Collaborative Business Processes* (CBPs) [4]. A collaborative business process (also called *process choreography* [5-6]) specifies the global view of interactions between organizations to achieve common business goals and serves as a contractual basis for the collaboration. Thus, the implementation of collaborative networks requires organizations can carry out the stages of the *Business Process Management* (BPM) lifecycle [5] to the agreed CBPs.

In the *analysis* and *design* stages, organizations have to define not only CBPs but also the inter-organizational *collaborations* in which they agree the CBPs to be executed. CBPs are abstract processes in the sense they are not directly executable but through a decentralized management, which implies the enactment of *Integration Business Processes* (IBPs) for each involved organization [7]. An integration business process (also known as *orchestration process* [5] or *public process* [6]) defines the public and private activities the organization has to perform to fulfill the message exchange agreed in the CBP. Thus organizations have also to define their IBPs from each agreed CBP in which they are involved.

The next stage is *implementation*, which consists in the development, configuration, and deployment of *Process-Aware Information Systems* (PAISs) required for each organization to execute their IBPs. Inter-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 each other to achieve the message exchange agreed in CBPs [7].

The *execution* stage consists in the execution 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 also 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.

All the abovementioned stages of the CBP management require to deal with: organization autonomy, decentralization, global view of message exchange, peer-to-peer interactions, and the use of suitable abstractions to represent communications [4]. Hence, a platform for CBPs should provide: services for defining collaborations along with the CBPs to execute; services to allow the definition of IBPs which should be consistent with the defined CBPs; services for generating implementations of PAISs and enacting IBPs of each organization in an autonomous way –so interactions among the organizations' PAISs are carried out in a decentralized way–; and services for CBPs monitoring.

Platforms for CPBs based on Internet technologies like Web services require each organization to develop, implement, and maintain PAISs using its own resources and infrastructure (hardware, software, network, etc.) [7, 24]. This increases complexity and costs for the organization.

- *D. Cocconi is with the Universidad Tecnológica Nacional (UTN) Facultad Regional San Francisco, Av. de la Universidad 501, 2400, San Francisco, Córdoba, Argentina. E-mail: dcocconi@sanfrancisco.utn.edu.ar*
- *J. Roa & P. Villarreal are with the CIDISI, UTN Fac. Reg. Santa Fe, Lavaise 610, 3000, Santa Fe, Argentina. E-mails: {pvillarr, jroa}@frsf.utn.edu.ar.*

978-1-5386-3057-0/17/$31.00 ©2017 IEEE Published by the IEEE Computer Society

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 [8]. So it is important to use technologies that can make it feasible for organizations like these the application of CBP management in collaborative networks.

Furthermore, existing proposals do not focus in dynamic aspects of inter-organizational collaborations [9], so they do not provide services that allow organizations to generate, deploy, and enact PAISs on-demand, accordingly to the CBPs that the organizations agree to carry out. These 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 CBP management 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; 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 management of CBPs in collaborative networks. The platform allows organizations to: (1) reduce costs and complexity by hiding the required infrastructure to organizations; (2) create and manage collaborations in an agile way, i.e. by generating and developing on-demand the IBPs and PAISs required to implement the agreed CBPs; (3) execute on-demand the CBPs; (4) monitor the CBPs by providing a common and shared view of the states of processes; (5) fulfill the issues of decentralized management and autonomy of the organizations for all of these services.

The proposed architecture of the cloud-based platform is also oriented to fulfill non-functional requirements of cloud services, in particular *elasticity*, *privacy*, and *portability*. Elasticity, at a process level, is important for improving performance and reducing costs of the cloud services for generating PAISs and executing processes. The platform can be deployed in a public cloud (managed by the infrastructure of a third party) and its services can be consumed by any of the organizations that join. However, to deal with privacy issues related to sensible information shared in a collaboration, or internal information managed by IBPs, organizations can deploy the platform in a private cloud with their own infrastructure and interoperate with the public cloud or other private clouds to manage CBPs with the other organizations. Portability is another important issue, considering that the platform is conceived to be independent of the cloud provider. Besides the public cloud, this is particularly useful for private cloud users, which can mount their own clouds using their infrastructure following recommended standards or freely decide the PaaS provider, as long as this one counts with the implementation of the standars in any way.

This work is organized as follows. Section 2 presents related work about business processes in the cloud. Section 3 describes the proposed solution and the main functionality provided by the platform through a use scenario. Section 4 describes the defined architecture for the platform. Finally, Section 5 presents conclusions and future works.

2 Related Work

Cloud computing is a new paradigm for creating distributed systems based on the Internet [10]. 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, reachable from a Web browser. 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 [11].

Cloud computing has several *service models*: 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 pretty mature today and there are several cloud applications available [8]. The PaaS model offers development services for building cloud applications. IaaS model implies the provisioning of hardware resources via virtualization [11]. Cloud computing also has several *deployment models*: *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 [12].

In the BPM domain, *Business Process as a Service* (BPaaS) is a new type of SaaS where processes can be defined, deployed, executed, and accessed over the Internet [13]. Existing approaches for offering BPM in cloud environments focus on fulfilling elasticity [14-16]. In [14] the solution seems to be very satisfying because it considers one pursue requirement: decentralization of elasticity control and the idea of generic strategies for dealing with elasticity. In [17] is described an approach for BPaaS considering elasticity and costs in an equally way, but for multi-tier web applications. In [18] authors propose a PaaS model for the design of cloud applications 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 [19]. However, the above works are not oriented to support CBPs.

There are few proposals that support cloud services for

CBPs [20-22]. These proposals have still important shortcomings for collaborative networks: (1) they provide a centralized approach for CBPs, been their execution driven by one organization, and they do not deal with autonomy and privacy issues; and (2) they just offer SaaS models to execute processes as services, but dynamic and agile collaborative networks also require a PaaS model for implementing PAISs on-demand –when organizations agree on managing a new CBP– or creating a new version of a CBP because they want to improve it.

Finally, outside the area of cloud computing, there are also software agent-based platforms proposed to execute CBPs [23-24]. In particular, in [24] it is proposed a platform that deals with the issues of dynamic collaborations. However, these proposals require organizations to deploy PAISs in their own private infrastructures, which relies in complexity, costs, and poor agility for managing collaborations, which results in a more difficult adoption of these solutions by the organizations that are interested in the implementation of collaborative networks.

3 Usage Scenario and Functionalities of the Platform

In this section we describe the proposed platform for CBP management. The purpose of the platform is to provide services to support the design, implementation, and execution stages of the collaborations and CBPs lifecycles. To describe the services and functionalities of the proposed platform we use a collaboration scenario from the domain of supply chain of the electronic industry, where three organizations (*Org. A*, *Org. B*, and *Org. C*), which perform the role of *Supplier*, *Distributor*, and *Retailer* respectively, want to implement a known collaborative model called CPFR (*Collaborative Planning, Forecasting, and Replenishment*). As part of this model, the organizations require to implement and execute the CBP called *Collaborative Order Management* (COM).

3.1 Cloud Services for the Design and Implementation of CBPs

For the mentioned scenario, Fig. 1.a shows the entities that are managed by the *cloud-services* that are provided by the platform for the design and implementation stages of CBPs. First, organizations are registred in the *cloud platform* and grouped in a *collaborative network* named *Electronic industry supply chain* (step 1). According to the stated scenario, by using a cloud service, organizations define a new *collaboration* named CPFR where they define goals and agreements to collaborate. An organization can create the collaboration and request to the rest of the organizations for joining it (step 2). In this scenario, the *Supplier*, *Distributor* and *Customer* join out the CPFR collaboration.

Then, 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 we developed in a previous work [7]. This repository can be accessed by the organizations that are in the same collaborative network by using cloud services provided

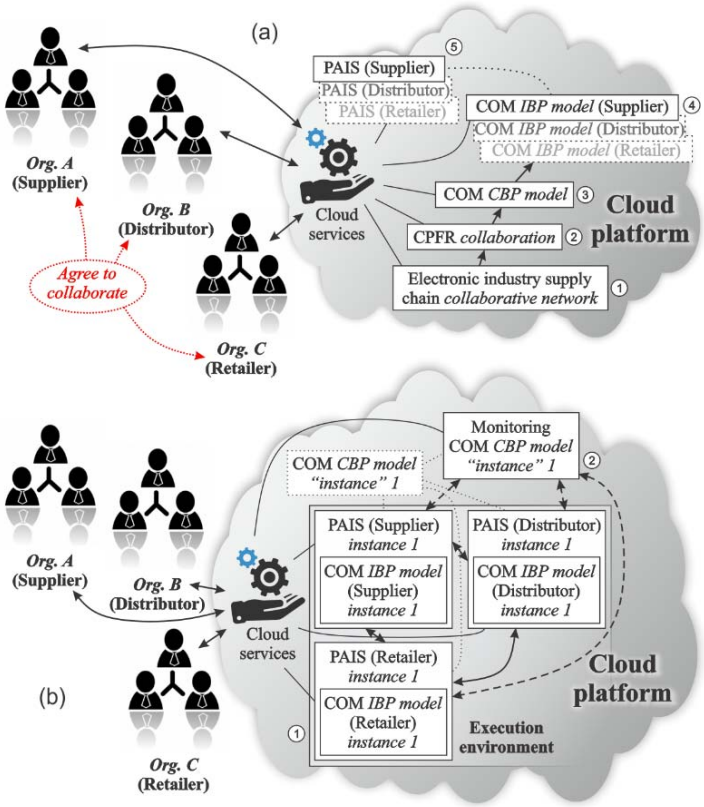

Fig. 1. Functionalities of the cloud platform for CBPs. (a) Design time. (b) Run-time.

by the platform. Through these services, organizations can select, update, and propose a *CBP model* or to create a new one according to their needs, and share it as part of the collaboration. In the scenario, organizations agree on to manage a CBP named COM (step 3). Therefore, a model of this process is stored in the CBPs repository and shared among the organizations. In order to guarantee correctness of the behavior of CBPs, the platform also provides services to perform the verification of CBP models by using the method proposed in [25].

Once a CBP model is agreed by the organizations, the platform provides cloud-based services to generate the IBPs that each organization needs for the CBP execution (step 4). To support this, the platform implements a model-driven method and tool [26] for generating the IBP models of each organization from the agreed CBP model. Since organizations may not want to share their private processes, an IBP repository is provided for each organization, in order to have access to its own IBP models only. This guarantees the autonomy of the organizations into the collaboration. In the scenario, an *IBP model* named COM (the abbreviation of the name of the CBP) is generated for the *Supplier*, the *Distributor*, and also the *Retailer*. Each of the IBP models contains the activities required to execute the collaboration just from the point of view of the corresponding organization. Thus, each organization is responsible of their private and public activities, meanwhile preserving the private information.

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 [27] to guarantee their correctness.

Then each organization can make use of cloud-based services to configure its IBP model with execution details –such as how to access Web services or databases–, to become it an executable IBP model and generate the PAIS that implements this process (step 5). The generated PAISs are also stored in a private repository for each organization, so organizations can manage its own PAISs to carry out collaborations. In the scenario, a *PAIS* is generated for the *Supplier*, *Distributor*, and *Retailer*, configured with the corresponding IBP to be executed by the organization.

3.2 Cloud Services for the Execution and Monitoring of CBPs

Once all PAISs are deployed, organizations can make use of cloud-services to support the execution of the CBPs, which is carried by means of the IBP executable models and the PAISs of each organization. Thus, the platform enables the execution of the CBP through the distributed (autonomous) execution of IBP instances supported by means of the generated PAISs (Fig. 1.b). This way, organizations are able to exchange information with each other in a peer-to-peer way, and share the information necessary to support 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, which enacts an instance of the corresponding IBP model (step 1). In the scenario, the PAIS of the *Supplier* organization is deployed and it instantiates a process instance of the IBP model, which is in charge of starting the process. In a similar way, the *Distributor* and *Retailer* request the deployment of their PAISs to instantiate the rest of the IBPs required to execute the CBP.

Once all PAISs are deployed and IBPs are instantiated, the *monitoring service* is started (step 2). This service enables that the organizations can be aware of the global state of the collaboration and, in particular, of the COM process. This allows the constant evaluation of the fulfillment of the established agreements, or to predict the quality of the collaboration with a given partner in the future [28].

4 Reference Architecture of the Platform

This section describes the architecture of the platform along with the components and interactions for each of the cloud services provided that support the functionalities explained in previous section. A general view of the architecture is shown in Fig. 2.

Although organizations would like to participate in collaborations by using a public cloud (as in the scenario described in previous section), sometimes they don't want to rely on a public service to manage sensitive information such as the involved in the private IBP models. To fulfill this privacy issue of collaborative networks, organizations can make use of a private cloud, which implies the deployment of the platform in their own IT infrastructures. This enables more autonomy to the organizations, and attends the requirement that some of them would want to maintain and preserve their private information and activities of their IBP processes.

Hence, the architecture of the platform enables that part of it can be deployed in a *public cloud* or in a *private cloud*. The public cloud contains the services for managing the collaborative networks, collaborations, and CBP model repository –i.e. all the entities and information shared by the organizations. For the services related to the management of the private context of each organization –i.e. the IBPs and the PAISs–, both public and private clouds provide the same services, however the place where information is shared and stored is different. By using the public cloud, all information and processes of an organization are stored and managed over the IT infrastructure of a third-party service provider. Instead, the use of a private cloud by an organization requires the deployment of the platform in their own IT infrastructures (or a third-party too).

Organizations interoperate with the cloud platform by using *Web applications*. In the public cloud, organizations access the platform on-demand via a Web browser, without relying in an IT infrastructure and paying only for the use of the service. From a logic perspective, all the private information of each organization is preserved to be exposed to others into the public cloud. This assures that only the owner organization can access its assigned *context*, preserving the privacy of all its process repository and sensitive information.

The architecture of the platform follows a two-tier model which consists of an *application layer* and a *service layer*. The *application layer* is comprised of Web applications that provide the user interfaces, so organizations can have access to the platform services. The *service layer* provides all cloud services required for the management of business process models involved in the collaboration.

Though, services of the cloud platform can be consumed: (1) from the *application layer* via *Web applications* that offer a front-end; or (2) by external applications that make use of the service APIs offered by components of the *service layer*. Organizations might even develop their own front-end using these service APIs.

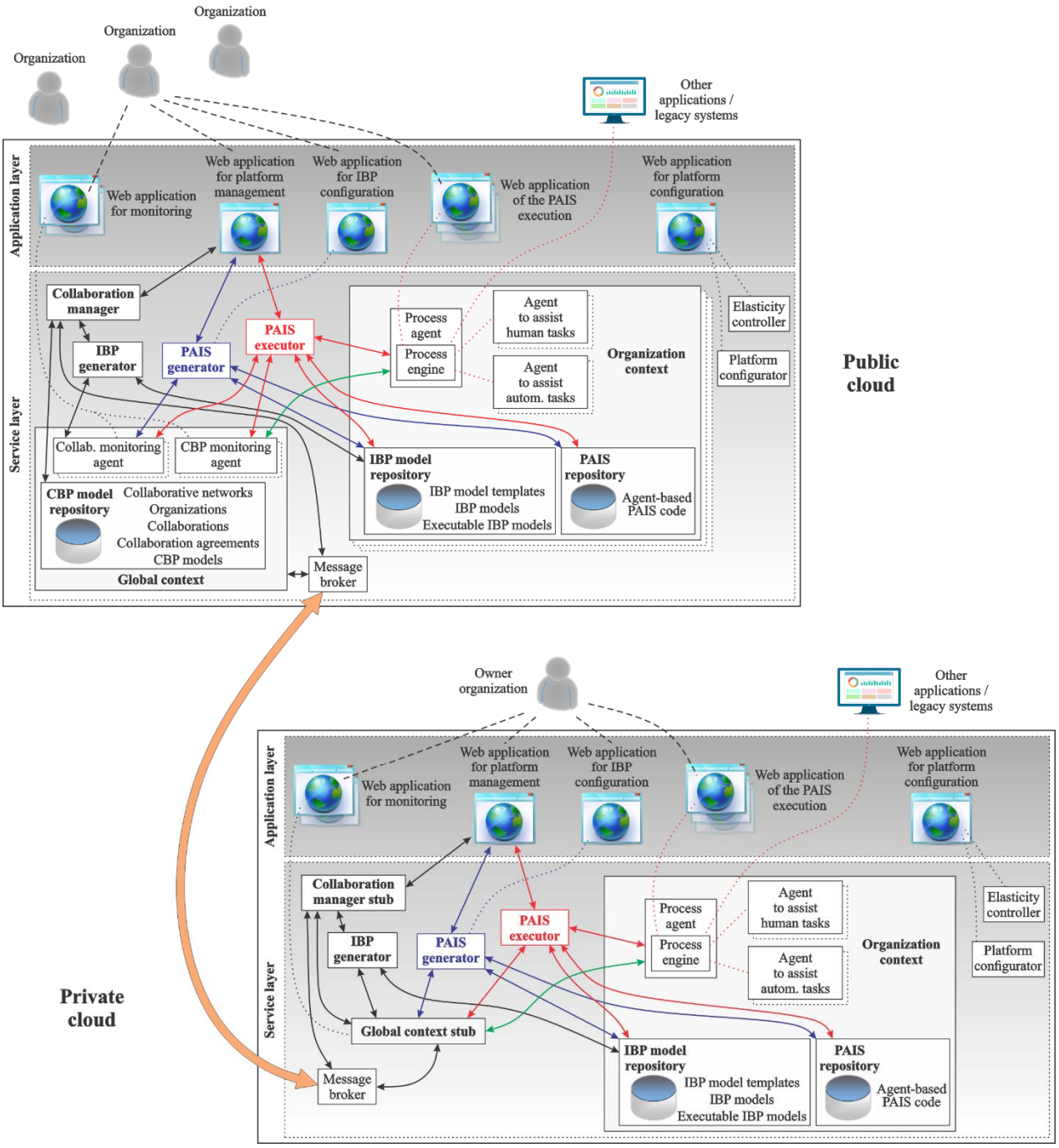

Fig. 2. Architecture of the cloud-based platform for CBP management.

The architecture has been designed to be fully cloud-aware component-based. On the one hand, some of these components are active, in the sense they can be executed autonomously by the organizations and share information in a peer-to-peer way (the PAISs and the monitoring components). Therefore, it is proposed to define these components in terms of software agents.

Considering elasticity from the point of view of components (and multi-tenancy in consequence) –as most of the processing tasks and the requests of organizations are materialized via software agents–, multi-tenancy is reflected to provide agent instances without degrading the performance of the platform. Then, the elasticity controller, which is implemented in both clouds, has to deal with the *agent level*, so the agent instances have to be *elastic* in such way. Considering elasticity at the agent level means that the platform implicitly provides elasticity at the *business process level*, because each agent represents a PAIS that executes the IBP of one organization, as it is described in Section 4.1. Elasticity controller takes into account service costs for ensuring a good cost-performance ratio to the organizations for implementing collaborative networks through this platform in a public cloud.

Portability (and vendor lock-in in consequence) is an important issue, considering that the platform is conceived to be independent of the cloud provider. Then, all

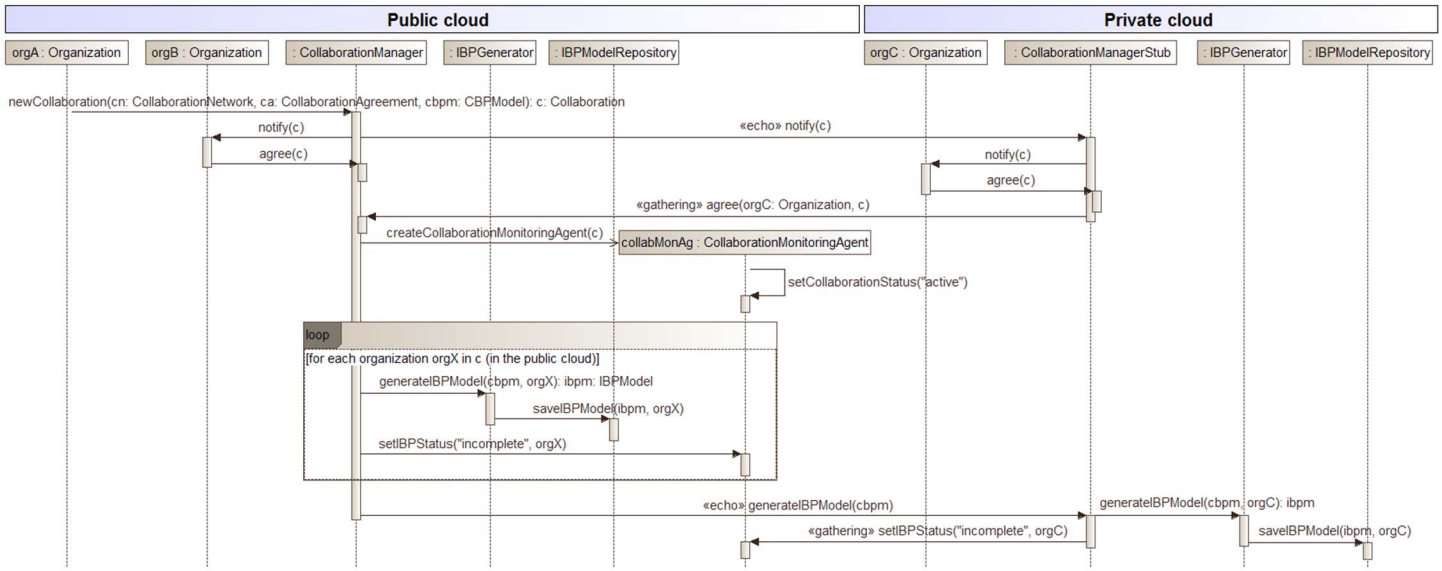

Fig. 3. Beginning of a collaboration and generation of the initial IBP models.

development and execution environments of the platform components are expressed via the promoted standard *OCCI-PaaS* [29] and the implementation for interaction with the different cloud providers (or platforms for cloud computing) is achieved via the *rOCCI* project [30] –to ensure portable cloud solutions. Besides the public cloud, this is particularly useful for private cloud users, because they can freely decide the PaaS 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* is available.

Following subsections describe the components of the architecture and their interactions, and finally the technologies used to implement the platform.

4.1 Components of the Architecture

A collaboration can be started at any moment by organizations that take part in a collaborative network. Fig. 3 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 using the public services. The right part of the figure shows the components allocated in the private cloud, owned by *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 of the private clouds, via the *message broker*. This negotiation process can derive in different scenarios, since the 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 stub; then, the 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 «*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*. Having all organizations agreed the collaboration, the *collaboration manager* creates a *collaboration monitoring agent* (*collabMonAg*) to take care of the status of the collaboration. This kind of agent only exists in public cloud and communicates via the *message broker* with the components of private clouds. The collaboration status is automatically set to "*active*" by the method *setCollaborationStatus*.

Next, for each organization involved in the collaboration, the *collaboration manager* sends a *generateIBPModel* message to the *IBP generator*, which creates an *IBP model* for each organization. To do that, the *IBP generator* employs the method and tool proposed in [26]. These generated models are saved in the *IBP model repository* of each organization via the *saveIBPModel* method.

The *collaboration monitoring agent* (*collabMonAg*) needs to be aware of the status of each IBP in order to elaborate the global status of the CBP, so the *collaboration manager* sends a *setIBPStatus* message specifying the state "*incomplete*" to *collabMonAg*.

For allowing organizations to configure their incomplete IBP models with all the information and resource links necessary to become a fully *executable IBP model*, the platform offers the services of the *PAIS generator* to perform this task, by invoking the *configureIBP* method (Fig. 4). Then, the *PAIS generator* gets the *IBP model* from the *IBP model repository* and the organization is able to apply the modifications necessary to complete the model via the *Web application for IBP configuration*. Once available the information to complete the *IBP model*, an *executable IBP model* is generated invoking the *generateExecutableIBPMod-*

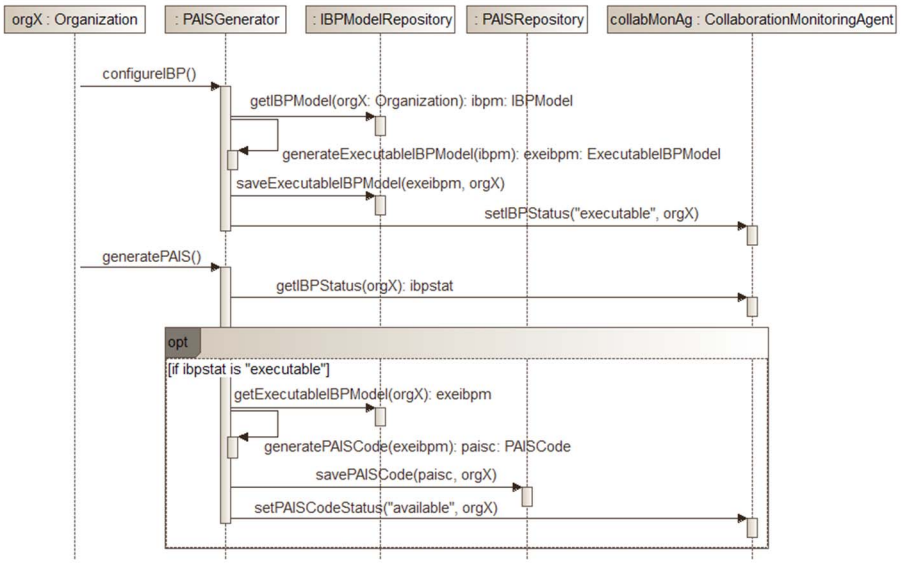

Fig. 4. Configuration of IBP models and generation of the code of the PAISs.

el method. This model can be directly interpreted by a PAIS and is saved into the *PAIS repository* using the *saveExecutableIBPModel* method. This situation is informed to *collabMonAg*, setting the IBP status to "*executable*".

From now on, each organization is able to generate the 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 generator* first verifies if the *executable IBP model* exists (it checks if the IBP status is "*executable*"). Next, it gets the model and generates the *PAIS code* by invoking the *generatePAISCode* method. The code is saved in the *PAIS repository* by invoking the *savePAISCode* method. Again, this must be reported to *collabMonAg* by sending a message *setPAISCodeStatus* with value "*available*".

When 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 Fig. 5.

Organizations can make use of the *PAIS executor* to enact IBPs by invoking of the *enactIBP* method. The *PAIS executor* first verifies if the *PAIS code* is available. To do so, a message *getPAISCodeStatus* is sent to *collabMonAg*. The next step consists in retrieving the *PAIS code* from the

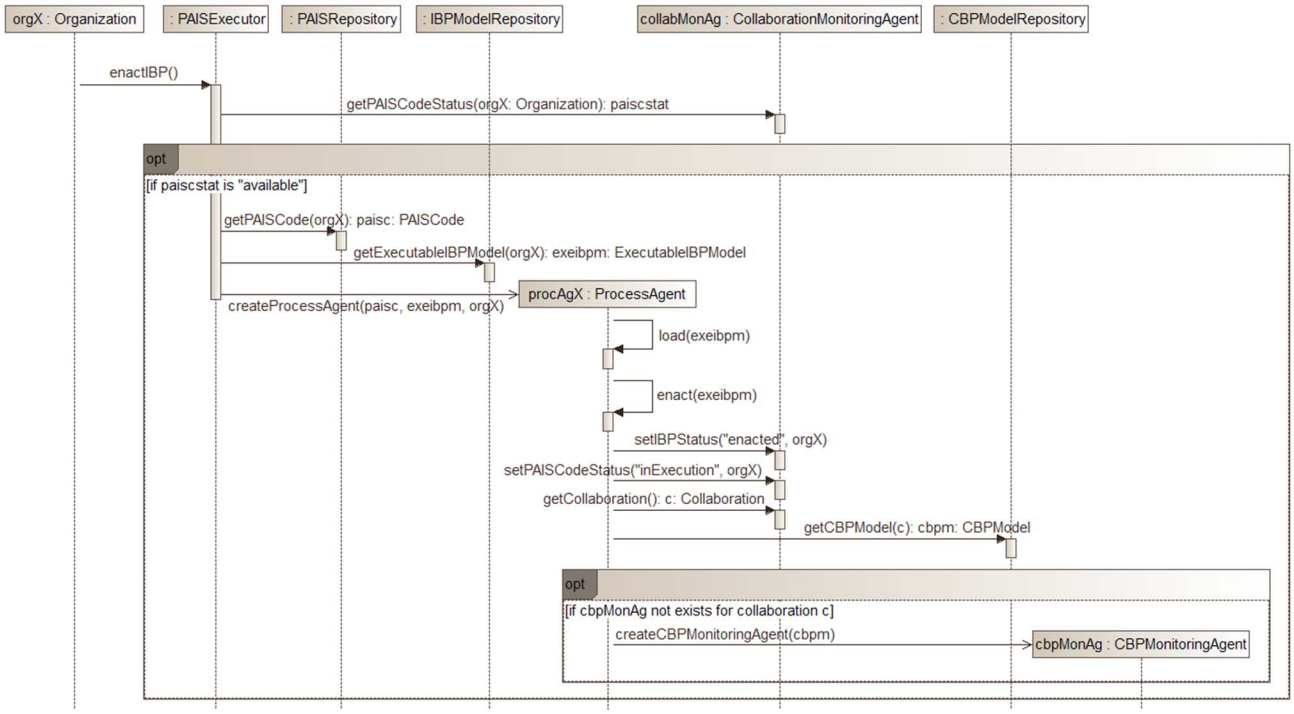

Fig. 5. Enactment of IBPs for CBP execution.

PAIS repository (method *getPAISCode*) and the *executable IBP model* from the *IBP model repository* (method *getExecutableIBPModel*). Next, a *process agent* (*procAgX*) is created by the *createProcessAgent* method. Once created, the *process agent* executes a sequence of tasks: it loads the *executable IBP model*; enacts that model; communicates with the *collaboration monitoring agent* to set the IBP status to "*enacted*" and the PAIS code status to "*inExecution*"; gets the current *collaboration* to obtain the CBP model; and finally, creates a software agent to monitor the execution of the CBP (if this agent was not previously created by the enactment of another IBP).

During the execution of a 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, auxiliar software agents are created: *agents to assist human tasks* and *agents to assist automated tasks*. The first ones interact with the organization by means of the *Web application of the PAIS execution*, which will show up the proper forms to complete information by a human user. The latter ones could interact with *external programs or legacy systems* or could require to invoke programs that implement decisions based on automation rules.

All public information referent to IBPs is updated by the *CBP monitoring agent*, so any of the organizations can follow the general status of the collaboration and check the information reported by the other organizations using the *Web application for monitoring*.

Each *process agent* in the collaboration expires once completed the execution of its IBP. Anyway, 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. Both kind of *monitoring agents* also expire when the CBP finishes.

4.2 Implementation of the Platform

Several technologies are used to implement the platform. The *Web applications* are developed using *JavaServer Pages* (JSP) and the offered APIs are implemented as *Web services*. The agents are performed 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.

The *platform configurator* and the *elasticity controller* are implemented as *Web services*, which make use of the *OCCI-PaaS standard* to define the resources of the cloud infrastructure and act as the API for interactions, and one of its popular implementations, the *rOCCI project*, to perform the link with cloud providers or cloud platforms.

5 Conclusions and Future Works

This work proposed a cloud-based platform to manage CBPs for dynamic and agile collaborative networks. The platform leverages a cloud PaaS model in order to provide services not only for the initiation of collaborations and execution of CBPs, but also for generating the applications required for this execution –i.e. the generation of the IBP models along with the PAISs of each organization. In this way, the platform allows supporting the main stages of the CBP management: design, implementation, and execution.

The defined platform architecture allows fulfilling the main functional requirements of the CBP management: organization autonomy, decentralization, global view of message exchange, and peer-to-peer interactions. This is achieved 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.

Privacy considerations have been taken into account during the design of the platform. This is the main reason why the platform offers a private cloud service; organizations can maintain control over all their sensitive information, internal processes, etc., and only have to provide public information only used to determine the global state of the collaboration. Security is not primarily an important concern because the platform is implemented by means of a PaaS model and most of the main threats in terms of security are dealt in lower levels of the development platform.

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

Portability is achieved through the configurator component and the elasticity controller, which are specified by making use of the OCCI-PaaS standard and the rOCCI implementation, enabling the construction of portable PaaS solutions. The configurator allows the deployment of the platform components over a cloud infrastructure, deciding the required resources for them. The elasticity controller increase or reduce the resources of the infrastructure to be used for each component, according to the organization needs. Thus, a third-party that wants to provide the platform can make use of any of the cloud providers supported by the rOCCI implementation. Also, for private clouds, organizations can implement the platform by using cloud platforms also supported by rOCCI. This allows organizations or third-party providers to select the underlying cloud infrastructure and decide it in terms of costs, elasticity, or other features, without dealing with the problems of dependent technologies.

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

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] 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.

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

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

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] 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.

[20] 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..

[21] Sprovieri, D., & Vogler, S. (2015, June). "Combining Business Processes and Cloud Services: A Marketplace for Processlets". In *International Conference on Business Information Systems* (pp. 247-259). Springer International Publishing.

[22] Muthusamy, V., & Jacobsen, H. A. (2010, September). "BPM in cloud architectures: Business process management with SLAs and events". In *International Conference on Business Process Management* (pp. 5-10). Springer Berlin Heidelberg.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

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

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

Diego Cocconi received his Engineer's degree in *Electrónica* (Electronics) at the UTN Fac. Reg. San Francisco in 2007. Currently, he is assisting to the last stage of the career *Ingeniería en Sistemas de Información* (Engineering, Information Systems) at the UTN Fac. Reg. San Francisco and started his PhD at the UTN Fac. Reg. Santa Fe. He is a professor of the subject *Administración de Recursos* (Resources Management) of the career *Ingeniería en Sistemas de Información* at the UTN Fac. Reg. San Francisco. His research interests include (Collaborative) Business Processes, Software Agents, Cloud Computing, Machine Learning, and Artificial Vision.

Jorge Roa is a full-time Professor of the Information System Department at the Santa Fe Regional Faculty, National Technological University, Santa Fe, Argentina. From 2014 to 2016 he was a postdoctoral research fellow of the Argentina's National Council of Scientific and Technical Research (CONICET).. He has been working in the Research and Development Center in Information Systems Engineering (CIDISI) since 2006. His research interests include Business Process Management, Service Oriented Computing, Formal Methods, and Artificial Intelligence. He received his PhD from the National Technological University, Argentina, and his degree in Information Systems Engineering from the same university.

Pablo Villarreal received a PhD degree in Information Systems Engineering from the National Technological University (UTN). He is a full-time Professor of the Information System Department at the UTN and an Adjunct Researcher at the CONICET. He is the head of the Center of Research and Development of Information System Engineering of the UTN. His main research and teaching interests include: information systems, business process management, workflow management, model-driven development, conceptual modeling, Business-to-Business collaborations, agent-based information systems, supply chain management. In these fields he has many published papers in journals and conferences. He serves as program committee member and reviewer at many conferences and has coordinated workshops and conferences. He has also coordinated more than 9 R+D projects.