Constructing hybrid architectures and dynamic services in Cloud BPM

José Martinez Garro
Universidad Nacional de La Plata
UNLP
La Plata, Argentina

Patricia Bazán
LINTI – Universidad Nacional de La Plata
UNLP
La Plata, Argentina

***Abstract*—along the last decade, organizations have developed BPM (Business Process Management) as a methodology to manage their own Business Processes, with a consequent evolution in the involved systems. Different technical issues such as high availability and growing connectivity, and also economical aspects generated by development and maintenance costs have guided a big number of organizations to choose a Cloud computing model. In this paradigm many economical and technical risks are diminished. Due to BPM being a naturally integrative model, there are concepts like workflow dynamism, or dynamic and mobile services that are reconsidered by entering into a cloud environment. In the present article we introduce a set of concepts presented in the current bibliography that are being rediscovered while BPM is inserted in the cloud, and we also implement proposals for most of them, in a particular BPMS (Business Process Management System) present in the market.**

Keywords — BPM; Cloud Computing; Mobile; SOA

I. Introduction

In this article we face the problem of including a Business Process Management System (BPMS) in a cloud oriented collaborative environment, with the particularity that it is an external environment to the organization. It is one of the purposes of this work to make a current bibliography analysis in sections II to IX, where we describe the different variants of a cloud model, its benefits and cons, hybrid architectures with embedded systems and the impact of using mobile devices. Then, from section X onwards we will make concrete contributions with some developments made using an internal BPMS. Using this tool we have designed solutions for several problems, like dynamic services and hybrid architectures, as well as the access to cloud based process engines from mobile devices. Finalizing the document we present some conclusions about the current state of the art and future work proposals in this research line.

II. Related Works

There are different trends in what comes to BPM in the cloud, but they are different if we are talking about research fields or trends in the market. Currently we can find research works tending to analyze the different paradigms of BPM (whether in the cloud or embedded), and how they escalate according to user's needs, connectivity that grows and mobile device incorporation. In [1], [2] and [3] especially we found trends like adaptive workflows, dynamic services and complex events. These references support the idea of the hybrid architecture and the dynamic services assigned by geolocation that we are proposing in further sections. Regarding the other references, we will cite each one of them in every related topic.

In relation to the commercial market, we find fewer advances than in the research area. Most of the available BPMS in the cloud are very similar to the embedded ones, and the concepts introduced in the present work and in [2] like decomposed processes or dynamic services are not present. The facts that we accomplished on our internal product are presented from subsection IX onwards. These features are not present in actual commercial products, showing again that Cloud BPM is a field in constant evolution, with a lot of work to be done.

III. BPM and Cloud

With the fast technological development in the context of application launching and execution using cloud based architectures, companies that began to choose this model are facing new problems. In particular, collaborative business processes with several interaction areas offer an optimization potential through the combination of cloud computing and BPM. A common factor between both paradigms is the flexible and agile approach. The cloud based computing model may be considered as an enabler for an improved combination of service oriented architectures, and also an agile procedure for Business Process Management. But this potential depends on the conditions imposed by the different frameworks, which can be viewed from technical and financial aspects.

A. Technical view:

From a technical point of view there are three dimensions in order to design, implement and successfully operate the different BPM tools in a cloud environment. These dimensions are: programming, integration and security.

- Programming: complex and distributed systems are easily reachable in current IT. In connection with obtaining more usability and flexibility, this complexity represents new requirements for Software Engineering. To solve this problem it is necessary to adopt new languages. So, based on new concepts and innovative techniques, the efforts invested in the development phase have been reduced to convert the complexity of these new aspects into a manageable element.
- Integration: this category can be divided in data integration, function integration and process integration. Under the light of the new challenges

involved, the current topic plays an important role in different scenarios. For example, a cloud based workflow can control distributed activities beyond the companies' border, mainly due to its easy accessibility. For a simplified execution of several process instances it is necessary to have integration interfaces and structured methods that allow joining the new components under the considered process.
- Security: this concept can be divided into three categories: functional security, information security and data security. All these categories have a significant relevance for BPM, especially in regard to business process grids and distributed process servers. Functional security specifies how the current status corresponds with the desired functionality status. The information security is focused in unauthorized changes or information extractions, as well as data security is in charge of the process related data.

Even more, from a technical point of view the question on "what processes are more appropriated to be executed in a cloud-based architecture" should be responded. The possible risks, such as insufficient integration options, location and integrity problems as well as programming interfaces should be taken in consideration.

B. *Financial view*

There are two dimensions from the financial point of view:
- Availability: the services provided by a cloud infrastructure can be accessed at any time because of the high availability model. Based in a high abstraction level, the customization and installation are significantly easier. In addition with this simplification, the final user is capable of working with the service immediately.
- Investment risk: in the context of the different variable billing models (for example "pay per transaction") the use of a cloud based service results in certain charges. These charges contain relevant costs given by transferences and transactions [1] [2] [3] [4].

IV. BENEFITS AND DRAWBACKS

Cloud based BPM provides users the possibility of using software in a "pay per use" way, instead of forcing them to make big investments in BPM software, hardware and maintenance, versus the traditional licensing applications. Systems can escalate up and down according to the user's needs. This means they do not have to worry about the over/ under resource provisioning because of the high adaptability provided currently by cloud service providers, as we can see in Figure 1.

The current model, on the other hand, has several low points. By putting a BPMS in the cloud, users may lose control over sensitive data. This aspect results major considering that business processes inside an organization may manage important information for it and its members. On the other hand, the non high computational activities' efficiency and effectiveness cannot be increased by putting them in the cloud, but rather these activities may get more expensive. For example, an activity which is not intensively computational could need to process a certain amount of data. The transference of these data to the cloud could take more time than the transmission to an embedded version installed locally. That transference could result bigger than the real necessity of processing. Even more, the cost of the activity may increase due the data transference. This element is one of the billing concepts in a cloud computing system because of the high connection availability [1] [2] [5] [6].

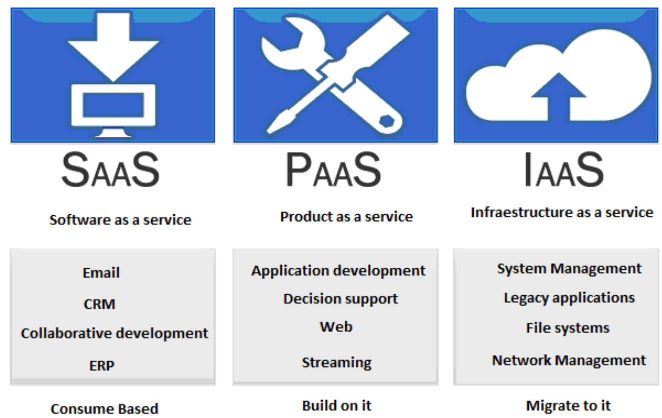

Fig. 1. Service model

V. SERVICE MODEL

A. *Infrastructure as a service (IaaS)*

When an application is moved to an IaaS model, the cloud user is responsible for the operating system, the middleware and the applications running on the virtual machine. The action of installing BPM software in an IaaS cloud solution is comparable to installing an embedded BPMS, since everything except the hardware is managed by the cloud user. Furthermore, the user has to make some security decisions in order to avoid intrusions. According to this, possible security measures are: port blocking, access control policies and updating the applications and the operating system frequently.

B. *Product as a Service (PaaS)*

By positioning a workflow based application in a model like "Product as a Service", the responsibilities for the user and the cloud provider are different. The execution engine is assumed as a part of the platform, so it is offered by the service provider. Users must upload their processes to run them in the cloud. The engine can be used by several users since the platform is shared. The responsibility for data storage and management is no longer in charge of the user, who has to deal with several security issues:

- The process models should not be readable by intruders in posession of a description file.
- Process models should not be altered by intruders.
- Process models should not be deployed in other servers.

In order to achieve these requirements, the process model descriptions should be encrypted and signed. The encryption ensures that process models are not readable by intruders. By the action of signing them, it can be assured that a file is only valid for a particular execution engine, and using it to point to another execution engine will provoke an error. This turns into utility considering that the same server can be accessed by different users in a shared environment.

Storing the application database can be an issue also. Data should be encrypted in order to not be readable by intruders. Data encryption in a relational database generates expressivity issues with queries using relational operators. For example, joins can have problems in an encrypted data context.

C. Software as a Service (SaaS)

By moving an application to a SaaS model, the cloud provider is now responsible for the application itself. The application is no longer an asset of the enterprise cloud user but it is offered by the cloud provider. The application may be given to multiple cloud users in a single or multiple tenant architecture. In a single tenant paradigm, an execution engine is installed for each process model. In a multi-tenant environment, multiple users and process models are served by a unique engine. The data stored by the cloud provider should be assured in order to prevent unintended accesses, both by the service provider or other users in the cloud. The same measures we have mentioned in the previous subsection related with signing and encryption can be applied to solving this problem.

In a multi-tenant architecture, different users access the same execution engine. The data used by one user should not be accessible to other cloud users. There are two possible solutions for this problem: in the first place, a database for each cloud user can be created. As an alternative, a column to each table where the user identifier is saved can be added. It is necessary to observe the scalability of both solutions: the amount of users could increase, and because of that, the need of resources too [1] [6] [7] [8].

VI. Combination Of Embedded And Cloud Schemas

Privacy protection is one of the barriers to execute BPM in a cloud environment. Not all users desire to put their sensitive data outside the organization. Besides, it is necessary to observe products portability and versions, and their availability in a cloud system. Another not minor problem is the efficiency.

The intensive computing activities may obtain benefits in the cloud due to the scalability and the computing force high availability. The non intensive computational tasks, on the other hand, not always take advantage of this context. The performance of one activity running in an embedded environment should be better than in the cloud because of the data that are transferred in order to execute the activity. These activities could also result expensive due to the fact that data transference is a billing criterion in the cloud.

- Architecture: in most BPM solutions, the process engine, the activities and the process data are located in the same side, even in an embedded or cloud solution. There are some papers introducing the PAD model (Process - Activity - Data) of Figure 2 as a distribution possibility for BPM in the cloud. In this approach, the process model, the involved activities and the data are separately distributed. The PAD model defines four possibilities of distribution:

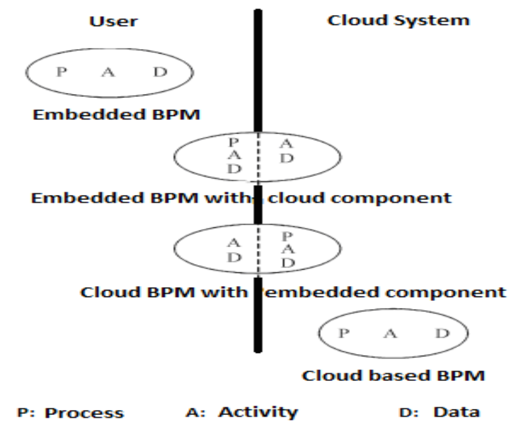

Fig. 2. PAD Distribution Schema [6]

1) The first pattern is the traditional alternative where all elements are distributed over the final user side.

2) The second pattern is useful when the user already has a BPMS, but the high computing activities are located in the cloud to increment their performance.

3) The third pattern is useful for the users who still do not have a BPMS, so they can use the cloud system in a "pay per use" way. In this approach the activities with low computing intensity or the ones with sensitive data management can be located on the final user side.

4) The fourth pattern is the cloud based model where all the elements are located in the cloud.

- Business processes consist of two kinds of flows: control and data. Control flows regulate the execution of activities and their sequence, while data flows determine how the information is transferred from one activity to the other inside the process. BPM engines must deal with the control of both kinds of flows. A data flow could contain sensitive data, so when a BPMS is deployed in the cloud, the content of those flows should be protected.
- An example of the proposed architecture could be a scenario where the engine in the cloud only deals with data flows using reference identifiers instead of real data. When an activity needs sensitive data, the data being transferred to the activity are managed under user supervision in an encryption tunnel. Sensitive data are saved in the final user side, and non sensitive data are saved in the cloud. This schema allows that sensitive data do not travel indiscriminately through the web.
- Optimal distribution: the cloud system costs have been a study object in different articles. There are several formulas to calculate the optimal distribution of activities, since they can be located in the cloud or in an embedded system. The calculation takes in consideration time costs, monetary costs and privacy risk costs. By using these formulas, users can make cost estimations about deploying part of their applications in an embedded or cloud system alternatively [2] [5] [6] [9].

VII. Process Decomposition

It is possible to generalize the distribution and identify a fifth pattern where the process engine, the activities and data are deployed in the cloud and in the final user. This solution presents two potential benefits:

1) The process engine regulates control and data flows. One activity receives data from the process engine and after its execution the produced data are passed again to the process engine. Consider now a sequence of activities located in the cloud, while the process engine is deployed in the final user. Each activity uses data produced by the previous activity as an income. Data are not passed directly from one activity to the other but they are sent to the process engine first. Since data transference is one of the billing factors in this model this kind of situations could become more expensive when large amounts of data are transmitted between activities. To avoid this problem a process engine can be added to the cloud, in order to regulate the control and data flows between activities located inside it. When a sequence of activities is located in the cloud, data are regulated by the process engine in the cloud. This reduces the amount of data to be transmitted between the cloud and the embedded system.

2) When the cloud is not accessible, users can execute business processes in a complete way in the embedded system until the former is available again.

In order to run a single business process between two separated engines, it should be split into two individual processes. It could be convenient for the users to take a distribution list of the process and its activities. The process can be automatically transformed into two business processes, one in the cloud and the other in the embedded system. The communication between both systems can be described using a choreography language, like BPEL. Besides, the distribution list can be created automatically according to the optimal distribution formulas mentioned in subsection IV.

Business process monitoring is more complicated now, since the process has been divided into two or more parts. As a solution, a monitoring tool can be developed for the original process, through the combination of the individual process monitoring details.

A possible approach to manage the process decomposition is to identify its structure and semantics. When the control and data dependencies are identified, the consequences of moving some activities from the embedded system to the cloud and vice versa can be researched. When the activity distribution consequences are known, a transformation model can be created.

Then, a business process and a list with marks are used to create two separated processes, one for the cloud and another for the final user. Also, a choreography description can be generated in order to describe the communication between both processes using some standard language, like BPEL [6] [10].

VIII. Particularities Of BPMS In The Cloud

A. Multiple web services in the same cloud instance

To reduce the overhead caused by initiating instances and to avoid extra costs, some additional web services can be developed in the same cloud instance, if they are adaptable to the hardware. This optimization is mostly for workflows with different cloud tasks, which can be executed in line inside the same instance. Using this optimization, the existing static images of virtual machine will not be able to be initiated dynamically: it is necessary to install additional software during the instance's execution time. The installation can be made statically using for example a script executed via SSH. Through it, a remote program is initiated to publish the web service in the cloud instance by using parameterization.

B. Dynamic task assignment to cloud resources

The idea of making outsourcing only of individual parts of an application to the cloud can be extended with a dynamic task assignment. The selection process is similar to the transversal three phase model of cloud computing. In phase one (or discovering phase) the cloud service broker creates a table in a database which provides information about the secure properties offered by the cloud suppliers. Possible properties in this context are, for example, special hardware, best performance, lower price, price/performance rate and location in the cloud (for reliability reasons or data closeness, as well as sensitiveness). This table must always be updated in order to not become irrelevant.

In the coincidence obtaining phase, the cloud service broker compares the properties required by the task with the table of secure properties. The cloud service providers that ensure all the required properties are task potential owners.

In the authentication phase, the service broker selects the cheapest potential owner and assigns it as the current proprietary for each cloud task [6] [7] [11] [12] [23].

IX. The Challenge Of Dynamic Services In BPM

The advances in mobile systems and embedded computing have guided to a computational environment that considers personalization, mobility and real time computing as key elements. As a result, SOA (Service Oriented Architecture) has adopted new security methods in the service, and also in the management and execution process. For business processes, these adaptations represent challenges in terms of process modeling, development and execution. In order to execute BPM in these new environments we should see what the required changes are. This makes possible to support services, which commonly are growing, dynamic and decentralized.

The research on emerging architectures based on web services puts BPM as a key component in the development of the next generation in web applications. The use of business process engines and models allows application designers and business analysts to make focus in the system functionality without the necessity of considering wider technical architectures.

The use of web services for creating loosely coupled business processes leans on the orchestration in order to accomplish the functionality of the desired distributed application. Typically this is made by the combination of stable services in an embedded workflow, or even in a cloud based installation. Nevertheless, in dynamic environments, services are not frequently static. They usually change their status and characteristics; or even the environment could be unstable and the workflow should adapt dynamically to changes during execution.

The environments with emerging web services are adopting mobile devices increasingly, as well as personal data and embedded low level functions. This type of services stretches BPM current limits. The emerging needs of these computational environments generate complex demands in business processes. In a typical process oriented application, the change can be managed through adding loops to the process, or making changes using templates as a reference. In more dynamic environments the services have to be renegotiated and the workflows are altered during execution.

Changing the process model during execution (for example, by inserting a loop or a fork) is a frequent research aspect (without major implications in current BPMS market). Meanwhile, business process adaptation depending on environment or resource changes still continues as an unexplored topic.

On the other hand, the viability studies to include the current topic in a cloud based BPMS are still remaining. There are some positive factors, such as the error adaptation capacity, or even changes in the service location. Alternatively, one negative factor could be to overload the server with functionality that decrements performance in tasks with high data transference. Because of this, time and financial costs result higher. These factors are not really important in an embedded BPMS [2] [3] [13].

X. Emerging Changes In BPM

BPM and SOA can be considered as a perfect combination in terms of integration. BPM's goal is to allow process designers to create applications without worrying about the subjacent technical architecture. This fact has been possible due to the service oriented representation of computational resources.

Web services in SOA present common interfaces and also support communication standards for data exchange.

Inside the integrated domain of the company and other systems with web interface, the application of BPM using process engines to execute templates (in languages like BPEL) is a design trend that represents scalability and common technology for organizations.

A. Mobile services

Recently, the massive adoption of mobile devices has guided to the development of a new generation in web applications. For most organizations, these applications represent a new possibility for process improvement (Figure 3).

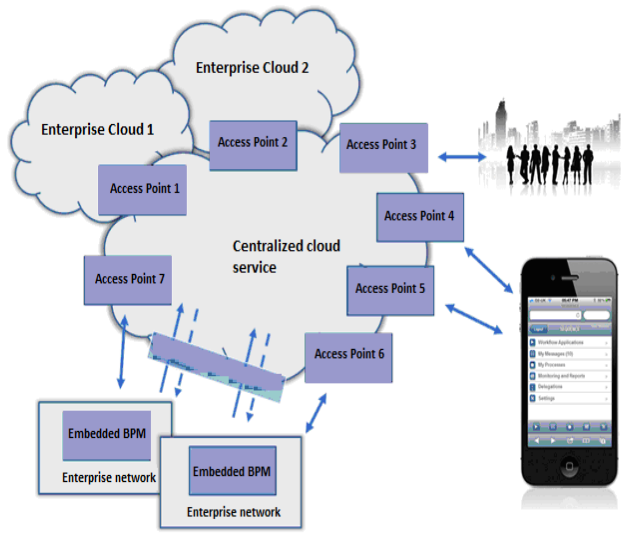

Fig. 3. Mobile Devices in a Cloud architecture

Some challenges for the business process designer can be observed when the workflow is present in a mobile device, or when some critical decisions have been made based on mobile services in an emerging environment.

In this context, workflows are present to handle the response to emergencies from a centralized perspective. Nevertheless, the change present in the central workflow is worsened by local factors. For example, the workflow could need to react to contextual information connected with ambulance and doctor's location in an emergency site. Here the device location is constantly changing and the context where they are could vary at anytime.

This non-predictable scenario cannot be modeled in a static workflow. BPM should adapt to the status of some services, and this status could be related to the context where the services actually are.

The inclusion of mobile devices gives more dynamism to the BPM model, even in an embedded environment. At the time of considering this inclusion in the cloud, there are some major aspects to take into account, such as sensitive data transferring. This should be done by using encryption and signing methods. The availability to access the cloud system from public networks is another important aspect, considering that it could be external to the enterprise network. We will delve into these aspects when we discuss the proposed implementing options [2] [13] [14].

B. Real time services

Different organizations (i.e. large scale manufacturers) have adopted BPM to control internal elements, such as the supplier chain management. The workflows around the office systems require minimum maintenance, and allow a quick integration with similar systems.

In the manufacturing domain, the inactivity time is a big concern and an increasing cost factor. Typical causes of this problem are machine failures or component supplying failures.

The control and integration with production lines using BPM mechanisms have some advantages in terms of reconfiguration and standardization. But there are some real problems with BPM in terms of security and reliability in the production line execution. These problems do not decrease in a cloud environment, though the control and monitoring from outside the organization are benefits of these approach. This generates more availability and accessibility in the monitoring system by considering a cloud context instead of an embedded one, which provides more flexibility for contingency treatment.

C. User Centered Services

Service providers lean over BPM to handle the different client choice ways through online systems. With standard workflows, user experience can be more controlled. This approach should join activities with web services. The identity management architectures use policy frameworks and trust rings in order to establish authentication and authorization rules between the service providers. These rules and mechanisms could be implemented through centralized workflows. The approaches centered in the user require additional contact, depending on the available security context. A possible solution for this problem is modeling the specific security interactions through annotations. These should be automatically executed using a process model transformation, commonly based on implementation and some BPMS special components.

D. User Centered Security

Reliability measurement and management are vital for service execution and data management inside the framework. The confidence awareness gives users the opportunity of expressing their needs in the network, and finding appropriated services. Typically the BPM operations are executed from centralized servers with predefined access rules. In a centered security environment this execution becomes more complex. The complexity grows due to the data associated with the policies. A big number of authentications and individual authorizations are needed in order to access to the distributed data object groups. This authentication reduces business process performance and could cause failures in applications.

In a cloud context these aspects have plenty of validity: sensitive data traffic and authentication in service execution become essential in shared environments. The possibilities of making controls in a distributed way are increased, and because of this the overhead produced by centralized authentication mechanisms is reduced [8] [9] [15] [16].

XI. IMPLEMENTATIONS CARRIED OUT

The different implementations we are introducing were developed using a BPMS implemented by our team. This product has been developed experimentally for an internal project, evolving then to a point where it was possible to put it. In a stable production context. From there it was possible to implement different researchable aspects, where we found the items presented in the current paper: hybrid architectures between cloud and embedded systems, besides the invocation of mobile and dynamic services. Table I introduces a short review of every aspect.

TABLE I. IMPLEMENTATION ASPECTS IN THE CLOUD

Aspect	Implementation
Dynamism in service assignment	In an embedded system, services in general are assigned to activities in design time, which makes their stability predictable. In a cloud scenario we find a large diversity of locations accessing to different points of the BPM service. Nowadays most of cloud servers allow implementing a service mirror model. This last feature grows by using mobile devices and, consequently more access points. For this reason the existence of routing tables in the execution nodes should be necessary. These tables allow according to certain criteria (such as client location, average response time, connection speed) deciding which executable version of the service is better to assign in order to improve performance. The process execution is more dynamic, and more robustness in error management is needed.
Process Decompositi on	The decomposed processes are located along the cloud and embedded systems, but they should be able to be executed and monitored as a unique entity. For this purpose, start and end message type events are used, where the finalization of one process step initiates the next one. In terms of monitoring, a web application and several web services are implemented in order to recompile the instance execution data along the architecture.
Mobile devices	The dynamism and adaptability of the BPMS are incremented, so the system utilization from dissimilar platforms is increased also. Reliable security standards are still observed as necessary, in order to enable the exchange of sensitive data between the clients and the server. This is more pronounced in a cloud environment, where the servers could be in a mirror model and synchronized in a distributed way.

XII. HYBRID SCHEMAS IMPLEMENTATION

The possibility of locating a BPMS in an external space to the organization (for example in a cloud computing architecture with a SaaS schema) makes feasible to access it from inside the organization through an Internet connection, as well as from any other external point. Considering this fact, besides the possibility of having clients accessing from mobile devices, the access points to the cloud are incremented.

This generates the following issues about process execution, and their corresponding proposed solutions:

- **Process Decomposition:** as exposed in Section V, the fact of putting a BPM server in the cloud generates the problem of what to do with sensitive data management. Facing this problem, this solution can be enounced: in case of publishing the corporative database (or at least part of it) in a cloud environment is not a viable choice according to the organizational security policies, the decomposition of the process is going to be necessary in order to implement a hybrid schema. In this scenario, the high computing activities can be located inside the cloud in order to take advantage of the computing performance, and the activities that make use of corporative sensitive data are located inside the organization in an embedded installation.
- **Decomposed process synchronization:** the disaggregated process is formally divided into sections according to the amount of participating servers.

According to this, it is going to be necessary to solve how to synchronize the servers in order to ensure the execution sequence of the process. There are, in theory, distinct ways to implement the synchronization, such as by messages or event monitoring. By using messages, the end event of each process part invokes the start of the next one. This can be made through start and end message type events, included both in the last version of standard BPMN (Business Process Management Notation), where the execution of the end event of a process throws a message to the BPMS in order to notify the finalization and require the execution of a process previously parameterized. This implementation was selected for our internal BPMS, where the notifications were implemented using a message queue and a daemon for pooling. This daemon receives messages and initiates instances of the required process. In this way, each intervenient server in this hybrid model (the embedded and also the cloud based ones) must have a copy of the pooling service in order to receive the finalization notifications and later notify the process engine. The result of this is to initiate the instances corresponding to the requested definition.

- **Decomposed process monitoring:** the biggest problem of having a process partitioned orientation is to monitor the different distributed instances, and at the same time to accomplish an integrated model of them under the optic of the "real process" which they belong to. In order to solve this inconvenient the following solution was designed: in first place it is necessary to associate the different instances with the original process, in order to recover them from the different existing servers. Once they are recovered, some kind of application in charge of gathering the recovered data and showing them seamlessly should be provided. The most important thing in this aspect is to accomplish monitoring transparency for the user, without forcing him to distinguish the server where each activity has been developed. This provides thus an integrated visualization of the different instances by seeing them as a same entity. The implementation of the current feature was made by a cloud resident web application. It was located there in order to access every intervenient server, whether cloud or embedded, and to ensure user access from any point. For this purpose it is important for the application to have a catalog with every existing server in the architecture, with their location information updated. Each intervening server has a copy of a web service which receives a process definition identifier and returns information about every existing instance associated with the sent definition. The returned information includes instance identification, current status (running, completed, suspended), current activity in case of non-completion status, start and end date. This way, the cloud resident web application sends an invocation of the web service with the selected process definition as a parameter to each intervening server, and receives the information of the associated instances. Then this information will be visualized in a web interface where the user can select a particular instance and observe its details. For this purpose the application comprises a web service to require each server the details of the associated activities. The information returned includes identification of the activity, associated participant, start date, current status and end date. After receiving this information the web application will allow the user to observe the activity's details transparently, without indicating the server information where they were executed. This helps to accomplish location transparency [6] [17] [18].

XIII. Dynamic Services Implementation

The fact of executing a distributed partitioned process allows us to reflect about the services that implement each activity and their location. In a hybrid model context, the organizations decide to maintain their embedded systems for privacy reasons related to the sensitive data, inter alia. Besides, we should consider the legacy applications used for implementing process activities: they could use corporative databases or non-scalable technologies, so it could be necessary to conserve them in an embedded context instead of locate them in the cloud. So we could execute activities in a cloud based BPMS, but really be invoking services located in another context, in this case, an embedded system, or even other independent cloud.

In this context the following problem appears: we have to consider the existence of multiple access points, potentiated besides by the amount of mobile devices trying to connect. Here we present the following solution strategy: service mirrors that allow optimizing the execution from different points of the architecture according to a convenient location.

According to the magnitude of processes and services, it could be necessary to consider cloud server mirrors, in order to decrement the bottle neck possibilities. This way, facing the necessity of executing a process activity and its consequent service, a geolocation service can be deployed in the server in order to decide which version of the process/service is more convenient for the user. It is possible to obtain the client's geolocation through different mechanisms automatically (such as the IP address in the case of a PC inserted in a network, the GPS or some QRCode scanning in the case of a mobile device) and then decide which is the most convenient node of the BPM hybrid system to execute the service requested by the client, as we can see in Figure 4.

For this purpose, BPM servers must have geolocation cross tables where, by giving a location, the hybrid system node to derive the client can be known.

The implementation can be as simple or as complex as a big the system is. If the nodes composing the system (between principal nodes and their mirrors) are relatively a few, it could be enough to include a routing table inside the BPM servers. This way, once obtained the device's location, the request can be derived automatically to the most convenient server.

Also, the same procedure could be applied in relation to the process or the service, according to the implementation criteria.

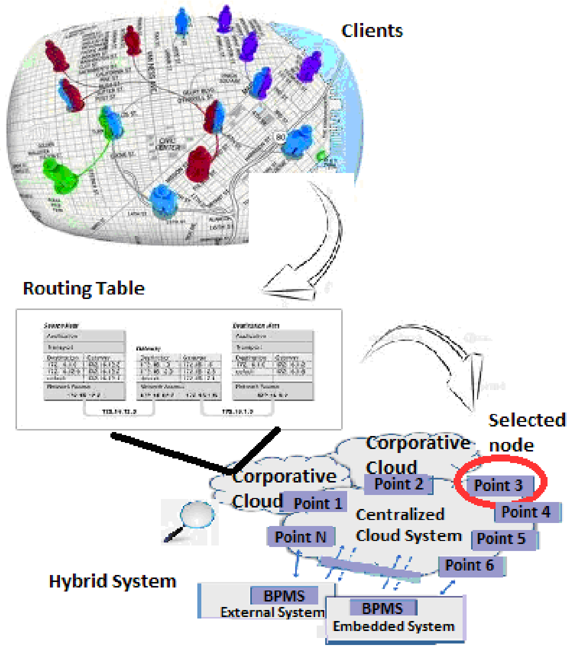

Fig. 4. Dynamic service selection using geolocation

In the case of our internal BPMS, we have developed only dynamic services due to the low amount of servers needed: only two embedded systems and a third one in a cloud environment. So, through a geolocation table that updates dynamically the node's IP address, we can decide which the most convenient service to invoke is, according to the device's location.

A possible scalable solution for a major system may be to implement a more complex geolocation system, such as the provided by open source systems like hostip.info [24]. This system has an easy-to-incorporate server side Rest API used for obtaining the client device's location. It can be used by incorporating extra parameters to the service invocation URL, and based on a more complex routing table, it is possible to decide by rules which the most convenient host for the redirection is. This dynamic feature, even when it improves the performance indicators by executing the most convenient services for the user considering availability and location, obligates the engine to be robust, since service execution is not predictable and errors must be handled on the fly [1] [19] [20].

XIV. Accessing The Cloud With Mobile Devices

Recently in BPM, the execution of process activities through mobile devices has gained ground. In the field of embedded BPMS, the use of these devices is increasingly common, and the exchange of sensitive data between them and the server is often considered. This problem has been solved through encryption algorithms and digital sign features with public and private keys, allowing sending sensitive data to the server using a wireless connection and then guaranteeing the sent data integrity and authenticity, as well as the sender information.

By considering a cloud environment instead of an embedded one, and changing essentially the technical scenario, the aspects considered previously for embedded systems require a revision to verify their validity. As a solution for the current problem, we enounce that cloud based BPMS will need sensitive data encryption and signing for the intervening devices, as well as the encryption of sensitive data conserved in the process server.

In this context we have developed a mobile device oriented application, based on Java and suitable for Android, which allows connecting with the process engine to visualize and complete the activity associated forms. It can also encrypt and sign data and then send them to the server.

The application mechanism is as follows: when it starts from the device, a connection with the server is opened using a login access. Once the client is authenticated the Rest API deployed on the server is used: particularly an operation which returns the completion-pending activities assigned to the user. Once this information has returned, the user can select any of the activities for completion. By selecting one of them, the associated form is displayed. Once the form's data are obtained, these are encrypted and signed using standards provided by Java. Between the most tested encryption algorithms we found DES and TripleDES, implemented by native libraries, and for digital signing we used JKS and PKCS12 as a mechanism to construct an internal certification authority. This way we could sign encrypted data by using asymmetric key certificates.

The use of mobile devices in a BPM context requires some security considerations before using the application. This should be done in order to guarantee integrity and authenticity in the sensitive data sent, as well as the authenticity and non-repudiation of the sender. For that purpose, users, as well as in other existing applications in the market, must previously register the device in the BPMS avoiding in this way any unauthorized access. Once the device is registered in the process server, the user can execute the application directly and connect with the server without generating an unauthorized access.

Once the access through mobile devices to the cloud is possible, the access points are incremented, in addition to the traditional existing nodes. In order to satisfy these accesses efficiently the duplication of adjustable services is necessary considering the user's location, as well as hybrid architectures guarantying portability and accessibility. It is also mandatory to analyze the security and integrity of the exchanged sensitive data [2] [8] [21] [22].

XV. Conclusions

As we have seen previously, BPM has incremented its action field along the last years. The market for traditional enterprise systems has suffered modifications in its licensing and commercialization paradigm, and BPM was not an exception for this, like other products of the same segment such as ERP or CRM. This adaptation pushes BPM to adapt existing technologies and methodologies according to new needs. Even when the cloud architecture solves the necessities of accessibility and high availability, there are some problems that should still be reviewed and researched deeply, such as sensitive data management in mobile devices, as well as in cloud server storing. For both cases we have provided implementations inside an internal BPMS, though we believe there is still a lack of real standardization inside the market regarding these aspects. These topics could generate more portability between products and installation schemas.

The system accessibility and the variety of devices are incremented, challenging the performance of the cloud system. This shows the necessity of mirror schemas and hybrid architectures, which allow to integrate existing applications without jeopardizing their integrity with other process areas that may require high computing benefits offered by the cloud architecture.

Even though the distribution between embedded and cloud systems was considered as a powerful alternative, it is very important to remember the relevance of the calculation formulas in order to estimate accurately the distribution costs, since the data transference is an issue to deal with. This subject is not a minor one, and the research on it is not really profound yet. Our focus is put on obtaining in the future an easy way to calculate if a task is a good or bad candidate to be located in the cloud, and also to obtain a certain methodology to optimize the data transference between the different distributed nodes in the architecture.

Nowadays our efforts are focused also in deepening the dynamic services model, optimizing the geolocation scenario. This paradigm takes relevance in a complex environment, where the distribution of the clients is a very decisive factor for service execution. When the distance between clients and servers is long, a good service selection and assignment is the difference between a good and a bad process performance.

On the other hand, considering the process lifecycle, the monitoring and improving phases are relatively new concepts in cloud BPM. Over this subject, our interest is to obtain a generic application that allows process mining applied to distributed processes.

The procedure will be based on decomposed processes over different instances of cloud architectures, and the challenge leans on obtaining a better version of the process using different algorithms and applying automatically they obtained improvements along the whole architecture.

REFERENCES

[1] T. Kirkham, S. Winfield, T. Haberecht, J. Müller, G. De Angelis, "The Challenge of Dynamic Services in Business Process Management", University of Nottingham, United Kingdom, Springer, 2011

[2] M. Minor, R. Bergmann, S. Görg, "Adaptive Workflow Management in the Cloud – Towards a Novel Platform as a Service", Business Information Systems II, University of Trier, Germany, 2012

[3] M Mevius, R. Stephan, P. Wiedmann, "Innovative Approach for Agile BPM", eKNOW 2013: The Fifth International Conference on Information, Process, and Knowledge Management, 2013.

[4] Dr. Manuel Goetz, "Integration of Business Process Management and Complex Event Processing", Germany, 2012.

[5] M. Gerhards, V. Sander, A. Belloum, "About the flexible Migration of Workflow Tasks to Clouds - Combining on and off premise Executions of Applications", CLOUD COMPUTING 2012: The Third International Conference on Cloud Computing, GRIDs, and Virtualization, 2012.

[6] Evert Duipmans, Dr. Luis Ferreira Pires, "Business Process Management in the cloud: Business Process as a Service (BPaaS)", University of Twente, April, 2012.

[7] S. Aleem, S. Molnar, and N. Mohamed, "Collaborative Business Process Modeling Approaches: A Review", In Proc. of the 2012 IEEE 21st International workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 274-279, June 2012.

[8] Dirk Fahland, Wil M.P. van der Aalst Eindhoven, "Simplifying Discovered Process Models in a Controlled Manner", University of Technology, The Netherlands, 2012.

[9] Hubert Scheuerlein, Falk Rauchfuss, Yves Dittmar, Rüdiger Molle, Torsten Lehmann, Nicole Pienkos, Utz Settmacher, "New methods for clinical pathways -Business Process Modeling Notation (BPMN) and Tangible Business Process Modeling (t.BPM)". Springer-Verlag 2012.

[10] Marielba Zacarias, Paula Ventura Martins, "Collaborative methods for Business Process Discovery", Portugal, Springer-Verlag 2012.

[11] Jiri Kolar, Tomas Pitner, "Agile BPM in the age of Cloud technologies", Scalable Computing: Practice and Experience, 2012.

[12] Andreas Lehmann and Dirk Fahland , "Information Flow Security for Business Process Models - just one click away", University of Rostock, Germany, 2012.

[13] Rafael Accorsi, Thomas Stocker, Günter Müller, "On the Exploitation of Process Mining for Security Audits: The Process Discovery Case", Department of Telematics, University of Freiburg, Germany, 2012.

[14] Aleš Frece, Gregor Srdić, Matjaž B. Jurič, "BPM and iBPMS in the Cloud", Proceedings of the 1st International Conference on Cloud Assisted ServiceS, Bled, 25 Octubre 2012

[15] Dr. Luis Ferreira Pires, "Business Process Management in the cloud: Business Process as a Service (BPaaS)", University of Twente, April, 2012.

[16] S Balzert, P Fettke, P Loos, "A Framework for Reflective Business Process Management", 45th Hawaii International Conference on System Sciences, USA, 2012.

[17] Marco Brambilla, Piero Fraternali, and Carmen Vaca, "BPMN and Design Patterns for Engineering Social BPM Solutions", Politecnico di Milano, Piazza L. da Vinci 32, Milano, Italy, 2012

[18] Marco Brambilla, Piero Fraternali, Carmen Vaca, Stefano Butti, "Combining Social Web and BPM for Improving Enterprise Performances: the BPM4People Approach to Social BPM", WWW 2012, European Projects Track, Abril 16–20, Lyon, France, 2012.

[19] S. Balzert, P. Fettke, P. Loos, "Enhancement of traditional Business Process Management with reflection – a new perspective for Organizational Learning", Institute for Information Systems (IWi) at German Research Center for Artificial Intelligence (DFKI), Germany, 2012.

[20] Huang Hua, Zhang Yi-Lai, Zhang Min, "A Survey of Cloud Workflow", Jingdezhen Ceramic Institute, Jingdezhen, Jiangxi, 333001, China, Proceedings of the 2nd International Conference On Systems Engineering and Modeling (ICSEM-13), 2013.

[21] Toàn Nguyên, Jean-Antoine-Désidéri, "Resilience Issues for Application Workflows on Clouds", Project OPALE, INRIA Grenoble Rhône-Alpes, ICNS 2012: The Eighth International Conference on Networking and Services, Grenoble, France, 2012.

[22] Markus D□ohring and Birgit Zimmermann, "vBPMN: Event-Aware Workflow Variants by Weaving BPMN2 and Business Rules", SAP Research, Darmstadt, Germany, 2011.

[23] Zhenyu Fang, Changqing Yin, "BPM Architecture Design Based on Cloud Computing", School of Software Engineering, Tongji University, Intelligent Information Management, Shanghai, China, 2010.

[24] HostIP.info, geolocation. Www.hostip.info. April, 2013.